

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the Communications of the ACM.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Mr. Donald B. Russell of the National Institute for Research in Nuclear Science, Chilton, England, has shown me the courtesy of sending me a copy of his letter on SLIP. I much appreciate the opportunity he has thus created for me to say a few words about the current state of the system and its various implementations.

Let me say first that all of his comments and observations are correct and proper. I would also want to add, however, that except for the errors pointed to in the published SLIP listing, the main point of his discussion is that FORTRAN is not an entirely machine-independent language. This is certainly true. Part of the attractiveness of SLIP is that it may be imbedded in any algebraic language which permits the calling of closed subroutines. It is of course obvious that SLIP is entirely host language-dependent. A FORTRAN-based SLIP will look quite different from a MAD-based SLIP, for example. A given host language is in general also dependent on the machine for which it is implemented and even, in some cases, on the monitor system within which that language is exercised. One must therefore look at the documentation of SLIP as represented by the paper [WEIZENBAUM, J. Symmetric list processor. Comm. ACM 6 (Sept. 1963), 524-544.] as a guide, not as a set of codes which can be taken over bodily and run blindly. In any event, a SLIP system which is intended for very wide and repeated use should have all its major functions (e.g. NUCELL, RCELL, LIST, NAMTST, the sequencing and advance operations, NEWTOP, POPTOP, NEWBOT, POPBOT, etc.) coded in assembly language. My own current version of SLIP which is MAD-based is so constructed and is therefore quite considerably more efficient than a similar but primitive-based system would be. The details of various implementations may differ considerably. But the general strategy of one SLIP system should closely resemble that of any other. This is not to say that any SLIP program should be executable under all SLIP implementations. After all, SLIP is not a free-standing language.

A more important point is that Mr. Russell is the fourth person to have gone to the trouble of implementing a SLIP system relying entirely on the original ACM publication, i.e. without any reference to any other SLIP implementer or to me. It happens that the Atlas people in London, England obtained a SLIP deck from me some time ago, a deck from which the errors pointed out by Mr. Russell had been removed. I don't know the fate of that deck. Perhaps no attempt was ever made to put SLIP on the London Atlas. In any case, I think Mr. Russell would have saved himself some trouble had he known of the parallel efforts of his countrymen. A number of people in this country have implemented SLIP for the 7090 family of machines, each stumbling over the difficulties introduced by the fact that the original SLIP listing was for the CDC 1604 machine and that it contains the errors pointed out by Mr. Russell.

To my knowledge SLIP currently exists on the 704-709-7090-7094 systems imbedded in FORTRAN II and in MAD, on the 7040 in FORTRAN, on the CDC 1604 and 3600 in FORTRAN '63, and on the Q32 in JOVIAL. There may be other implementations I know nothing about. It may well be a service to silent workers in far corners of the computing world if an inventory of SLIP systems were now to be taken and published. I volunteer, by means of this letter, to act as a clearing house for such SLIP inventory information for the next month or so. At the end of that time, I will submit the inventory to the *Communications of the ACM*.

As I have already indicated, I am currently maintaining my own SLIP system in MAD. In that system certain functions have been eliminated and certain new ones (particularly in the areas of input/output and text manipulation) added. In addition, I have an interpretive version of SLIP operating in the M. I. T. (MAC) timesharing environment. Any such additions and changes made by others should be included in the inventory. Communications should be addressed to me at the address given below.

May I offer my compliments to the several individuals who, like Mr. Russell, have struggled through my original manuscript with such great diligence and persistence as to find out how SLIP works.

Marvin Ratner

c/o Computer Science Dept. 311 North Park Street University of Wisconsin Madison Wisconsin

Dear Mr. Weizenbaum,

In response to your letter in the recent ACM Communications I wish to inform you as to the state of SLIP at the Univ. of Wisconsin and all errors that I have discovered x up to date in the system . As of new SlIP is on the Library tape on the 1604 and is being used ky (and taught) in two computer science courses . Next semester I will attempt to write a general manual on SLIP to supplement km the documentation in the Communications . If any work has been done in this vein I would appreciate information about saidk . To x put the SL IP system on the library tape we added dq to the end of each entry point in the system to avoid conflicts with xm users who might have subroutines with names such as "TOP ".

The following is a list of errors which I have found and corrected to date .

1. Function kxxxxxx LSSCPY ---- after line 559 add if(LNKR(W) - LPNTR(LR))x5x8x5 6, 7, 6 7 $\mathbf{X} = ADVLWR(LR, K)$ GO TO 5 IF(LNKR(W(20 - LPNTR(LR)) 5. 8. 5 6 X = ADVLWR (LR, K)8 The above takes care of the case when the list which is being copied happens to be either W(1) or W(2) . A similiar correction is needed for the subroutine LSTEQL when either of the lists being comparee is W or W(2) . 2. & Function LSSCPY ----- after line 534 add IF(LNKR(W(2) - LPNTR(LRA)) 41, 12, 41 IF(LNKR (W) - LPNTR(LRA)) 11, 12, 11 41 ¥ XA = ADVLWR (LRA, KA) 12 IF (LNKR (W) - LPNTR(LRB)) 42, 13, 42 11 if (LNKR(W(2) - LPNTR (LRB)) 8,13, 8 42 XB = ADVLWR (LRB, KB)13 Function X LSSCPY ---- After line 562 add 3. **EX** L = LNKR (CONT (W(2)))CALL STRIND (0, L+ 1) Call SETIND (0,-1,-1, L)

This **EXTREMENT** correction prevents the List copy which has just been created from being returned to available space when the top cell of W(2) which contains it in name format $\dot{x}xx$ and has been deleted comes up to the top of LAVS .

4. Function LVLRV1 ----- li ne 434 should x3 read IF(LNKR(CONT (LVLRV1 + 1 00) 2, 3, 2
5. function LVLRVT ----- Line 423 should read
IXXX I IF(LNKR (CONT (LVLRVT + 1))) 2,3,2

If you wish to correspond over the summer , please address said correspondence to Dr. Ralph Lomdon . Computing Science \mathbb{R} Dept. UNiv of Wisc. the faculty member under whom I am working **xm** as I shall not be here over the summer .

Yours tryly

marin Ratha

Marvin Tatner

P.S. errors continued

5. functions INLSTL AND INLSTR (M,N) ----- After lines 152 and złźć 166 add

IF (LISTMT (L)) 2.1.2

This coprects the error of putting a cell on the list with an ID of 2 when you insert an empty list onto the host list .

P.S.#2 Correction number 3 causes the created copy list to have a reference count of 1 rather than 0 by virtue of its being put on $M(2 \ W(2)$. This however seems more in line with k normal list creations by the programmer .

-3008 2 N MAY 21, 1966 LISTING OF 1620 SLIP SYSTEM DICK SITES DAVID P. KELLEHER LE CE+37++CE-11 CE CE+31 VIIVOCE 41030 91030 05 0138188312

T	ZZ LNKL 0630
the state of the s	7700
	LZSPS
	*LIST CARD
	*ASSEMBLE RELØCATABLE
_	*STØRE RELØADABLE
	*NAME LNKL
	*ID NUMBER 0630
	A1000**** LNKL 0630
	A1010Q DS ,*+101
	A1020 DC 6,98/898,5-Q
	A1030 DAC 6, LNKL ,7-Q
	A1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
12	A1050 DSC 17,0,0
6-1	A1060 DØRGQ-100
	A1070FAC DS ,2492
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A1080BETA DS ,2630
N	A1090FXZ D5 ,3099
2	A1100 DS 5
0 CK	AILIOENT AM *-1,5,10
st	ATTEC TE CETT
	AII30 BNF TF (CF+11
T	AII40CF CF CF+11
-Wai	A1150 TF CF+11,-CF-11
NIFO	A1160TF AM ENT-1,2,10
n-#	A1170 TFL BETA,-CF-11
	A1180 TF FAC,FXZ
-8	A1190 MA FAC,BETA-5
a francisco de la compañía de la com	A1200 B7 1-ENT
	A1210LENGTHDC 2,
	A1220 DEND
12	
11	
10	
9	
8	
7	
6	
5	
4	

-	77 INCD 0422
<u> </u>	ZZ LINKR U052
0	ZZSPS
	*LIST CARD
~~	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
	*NAME LNKR
N	*ID NUMBER 0632
	B1000**** LNKR 0632
	B1010Q DS ,*+101
	B1020 DC 6,987898,5-Q
•	B1030 DAC 6, LNKR ,7-Q
1	B1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	B1050 DSC 17,0,0
•	B1060 DØRGQ-100
	B1070FAC DS ,2492
ž.	B1080FXZ DS ,3099
• 2 <u> </u>	BI090 DS 5
¥0	BILOOENI AM *-1,5,10
ta	BIIIO TE CE+II, I+ENT
	BIIZO BNF TF,CF+II
T	
	DI140 IF (F+II)=(F+II) PIIEOTE AM ENT-1 2 10
	PIIGO TE EACEVZ
Ě	BILTO MA FACHCE-11
	BILLO B7 1-ENT
• •	B11901 ENGTHDC 2.1
A CARLES	B1200 DEND
	CTOSC DESMETABLESSETTION OF LINE PROVIDE TO THE SCHEREFORD OF DESCRIPTION OF THE SCHEREFORD OF THE SCH
Section States	C10100 DR. 744101
•	CIODOWANA ID NO.0034
10	ALDVNINGES DROV : INCLASE
	· · · · · · · · · · · · · · · · · · ·
	# 21GSE KEL GY OVER L
	WYSDEWHIC BULNCWIWHE
8	WEIWE CYED DEAD 1
N 7	
6	
5	
3	
2	

-	ZZ ID 0634
	ZZSPS
0	
	*LIST CARD
1	*ASSEMBLE RELØCATABLE
μO.	*STØRE RELØADABLE
	*NAME ID
	*ID NUMBER 0634
0	C1000**** ID 0634
	C1010Q DS ,*+101
-	C1020 DC 6,987898,5-Q
	C1030 DAC 6, ID ,7-Q
	C1040 DVLC22-Q+5+LENGTH+2+8+2+10+5+ENT-6+5+0+30+0
	C1050 DSC 17,0,0
	C1060 DØRGQ-100
	CIOTOFAC DS 92492
	C1080BETA DS ,2630
01	C1090FXZ DS ,3099
	CILICENT AM #=190910
0	
2	
-	
20	
	C1190 DNF #+249DETA=1 C1192 AM EAC-1-10
	C1194 BNE #424-BETA=2
	C1196 AM EAC-2-10
	C1200 B7 1-ENT
	C1210LENGTHDC 2.1
	C1220 DEND
1	
0	
	AV22FABLE BEIDEVIVELS
	FUTAL CARD
0	
	ST282
0	SS TWES DESS
0	

•	
-	ZZ CØNT 0636
	77 CD C
O	ZZGPG
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	ALLET CADD
N	*LIST CARD
	*STORE RELOCATABLE
	*NAME CONT
~	*ID NUMBER 0636
	D1000**** CØNT 0636
	D1010Q DS **+101
	D1020 DC 6,987898,5-Q
•	D1030 DAC 6.CONT .7-Q
	D1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	D1050 DSC 17,0,0
•	D1060 DØRGQ-100
	D1070FAC DS ,2492 DE ACTING DE ALEBANCA DE DITELLA ENLEA DE MARCA DE
N	D1080FXZ DS ,3099
6 i	D1082CALL BTM ,*+11
ť	D1084INITASDS ,*-5
s10	D1090 DS 5
	DIIOOENT B CALL,,,TRANSFER VECTØR
	D1105SLADR DS ,*-5
BW	D1109 AM ENT-1,5,10
	DIIIO TF CF+11,1-ENT
	D1120 BNF TF.CF+11
Care Care	DII30CF CF CF+11
8	D1140 TF CF+11,-CF-11
	DIIDUIF AM ENI-1,2,10
-	DIIOU MA *+30,-CF-II
	DIIO ME **209-CE-II DII70 RTAM-SLADD79-CALL TO INTEDNAL UTILITY DOUTINE
	DI175 TEL EAC 90
	D1180 B7 1-FNT
	D119000 DAC 6.INITAS.
12	D1192 DVLC+5+INITAS
•	D1194 DC 2**
10	D1196LENGTHDS ,QQ-1
9	D1200 DEND
•	
	STREAM WY THERE AND
5	
•	SARVE-ANALVE DEDUCTOR LEVALE AND
3	
2	

•	
	ZZ INITAS 0640
	ZZSPS
0	
	*LIST CARD
	*ASSEMBLE RELØCATABLE
9	*STØRE RELØADABLE
	*NAME INITAS
~	*ID NUMBER 0640
2	E1000**** INITAS 0640
~	
2	E1050 DAC 091N11A597-Q
1185	E1050 DCC 17-0-0
0.0	E1053 DØRGO-100
-	E1057HIGH DS +434
2	E1061CØMADDDS +2231
0	E1062FAC DS +2492
ž	E1063BETA DS .2630
510	E1064FXZ DS +3099
0	E1066FX1 DS ,3109
	E1070 DS 5
W	E1080ENT AM *-1,7,10
Leon	E1082 B7 *+8
ND-2	E1084 TFM *-2,NEXT
R1E	E1090**** LINK TRANSFER VECTØRS TØ EXTERNALLY USED UTILITY ENTRY PØINTS
0;	E1100 TRNM,SLADR.
and the second	E1110CØNT DS ,*-5
	EIIII TRNM,SLADR, ENGLASSES TO SERVER STORE CONTO
0	EIIIZSETINDDS **-5
1.1.1	EIII3 TRNM,SLADR.
	EIII4STRINDDS #=5
9	EIZZU**** CHAIN UP FREE SPACE - INITIALIZATION OF CHAINER
12	E1230 IF LWRLMAHIGH
- ¹¹	EICOD IF UPREMICOMADU
0 10	
9	F1240 TOM LWRIM.0
A 8	
7	E1261 CM UPRIM-2 10
6	E1262 BNE *+32
0 5	E1263 TEM MES NMS2
A	
3	
0	

	E1264	B7 ERRØR
	E1265	TF DISK+8,UPRLM-2
	E1267	TDM UPRLM-1
	E1270	TDM UPRLM
	E1280	A UPRLM, LWRLM
1999	E1283	AM LWRLM, 1, 10
	E1286	TF DISK+13+LWRLM
	E1289	AM LWRLM, 19, 10 LOSS DECOMPANY AND LONG BUILDED BUILDE BUILDED BUILDE BUILDED
	E1292	TE AVSLL AVSLR
	E1294	SM AVSIL \$5.10
	E1300****	PERFORM CHAINING - MAY BE RE-USED BY ENTERING AT XXX
	E1301****	AFTER SETTING BLKN AND RETN+6.
	E1310XXX	BTAMSLADR • 78
	E1312BLKN	DS **-4
	F1315	TE FAC+FX1
	F1316	TD FAC-4-BLKN
	F1317	CE EAC-4
-	E1310	
	E1310	MA -AVELL -AVELD
	E1320	TD MECITIO-BLKN
	E1320	WATYMECA
	E13801000	MATTMEDI
	E1300	AM 57910910
	E1000	
	E1410	AM 00-10-10
	E1410	AM 99910910
	E1430	
	E1440	AM TAVSLL 92910
	E1450Ever	C 99,UPRLM
	E1460	BNE LØØP
	E1462	SM 99,10,10
	E1464	
_	E1470	IF BEIA, -AVSLL
	E1472	SFA BETA-3,,7
	E1473TØTAL	DS ,*
_	E147400160	MM BETA, 5, 10
	E1476	A TØTAL,98
	E1480	TNF MES2+8, TØTAL
	E1485	MA FAC, TØTAL
	E1490	WATYMES2
	E1500	SM -AVSLL,2,10 AND
	E1505RETN	B7 1-ENT SOURCE = ALWARDS PETE-VOISE 22 18 A BELORME
	E1525NEXT	TF FAC,FXZ
	E1526***	ATTEMPT TØ LØCATE MØRE SPACE
	E1999	B7 1-ENT

17.78		
1		
	E2000**** E	IXTERNALLY CALLED UTILITIES
	E2011*	
	E2012**** C	ALLING SEQUENCE = BTAM, SLADR, SLIP-ADDRESS, 78 / RETURNS
•	E2013****	MACHINE ADDRESS IN 99 AFTER PØSSIBLY TURNING A DISK PAGE.
	E2020 D	DAC 3,
	E2022SLADR T	F 99.LWRLM
0	E2025 B	INC3NØTRP
	P	UT SWAP
	E2031NØTRP B	NE CHEKASLADR-1
2	E2032 D	
	E2035CHEV D	
-	EZOSSCHER B	SNV ADD
~	E2030 (	F BACK+10
- ~	EZO40ADD A	98,SLADR-I
- 92	E2045 B	V ERRI
	E2050 C	M SLADR-5,,1011
<u> </u>	E2060PAGE D	S ,*
101-1	E2070 B	E INCOR STO
N.	E2080**** B	LØCK NØT IN CØRE
P 0	E2290 H	12 12 10 · 10
×	E2400INCØR C	99.UPRLM
015	E2410 B	H ERR1
0	E2411BACK A	M *+8, 7910 TURN POSSIBLE CHECK BACK ON
	E2412 · S	F BACK+10
- Į -	F2413 B	B2
ORI -	F2415FRR1 T	NE NMS1+36-SLADR-1
ain -	F2420FRRØR D	CTV
- E	F2430 W	
- na	E2435MES D	
9.8	E2440 C	
-	E2440 C	ALLEXIT
	E3000MESI D	MES, A, PAGE 9 - (E)
	E3001MES2 D	MES, A, XXXXX SLIP CELLS INITIALIZED. (E)
	E3010NMS1 D	MES,A,PSEUDØ-ADDRESS XXXX ØUT ØF RANGE(E)
	E3015NMS2 D	MES,A,SLIP ØVERLAP(E)
)	E3120**** I	NTERNALLY REFERENCED CONSTANTS
12	E3130LWRLM D	S 5, ADDRESS ØF RIGHTMØST DIGIT ØF PSEUDØ-WØRD NUMBER (X0000)
	E3132UPRLM D	S 5, ADDRESS ØF LAST PSEUDØ-WØRD (END ØF SECTØR BLØCK)
	E3140DISK D	DA
10	E3145SWAP D	D .DISK
9	E3340AVSIL D	S 5. PRINTERS TO FIELD ADDRESSES OF PICHT AND LEFT
- ⁸	E3350AVSLD D	S ALWELMA LINKS OF AVEL IN CODE ISLID CELL VOCCOL
7_	E3440444	VTEDNALLY USED UTILITY ENTRY LINKAGES
6		ATERNALLT USED UTILITY ENTRY LINKAGES
5	E34DUSLADR.N	OP SLAUK
1 4 -	E3460 D	086*-4
з		
2		

0	
•	
-	
	E3690 DVLC+5+CØNT
2	E3691 DAC 6.SETIND.
	E3692 DVLC+5+SETIND
N	E3694 DAC 6,STRIND
•	E3696 DVLC,5,STRIND
	E3700 DC 2, 10 - 10
V	E3710LENGTHDS ,QQ-1
•	E3720 DEND
1. 1 1 <u>2 1 1 1</u>	EIZZO BAR ENTHICHTTA
•	EISODIE DEBRACENII
8	E1180 IF CF+11++CF+11
	ETTROCE CE CETT
•	
ORM	
¥	
100	
6 ····	
	ETTODCETT DEV
· · · · · · · · · · · · · · · · · · ·	Eloadruka dav
LEOR	LICHOFUKE DZV
NIN-2	
RIER	FIDSBFAC DS +2492
S S	
	L1050 DSC 17+0+0
	F1040 DVLC22+0.5.LENGTH.2.8.2.10*5.ENTHA45.0.30.0
•	FIOND DAT A STIDIR V-0
1	
-	
-	
12	SUBJEC DELIETY
A	#219KE KEFBYGYBLE
10	*VEREWELE RELEVENEE
9	WI 121 CVOD
6	Sacha and an and an
5	
• 4	
3	
2	
•	

1.1	
F	ZZ SETDIR 0644
1	77000
1	ZZSPS
	ALLET CARD
	*LIST CARD
	*ASSEMBLE RELOCATABLE
	*STORE RELOADABLE
	*ID NUMBER 0644
	F1000**** SETDIR 0644
	F10100 DS ++101
-	F1020 DC 6-987898-5-0
	F1030 DAC 6.SETDLP.7-0
	F1040 DVL C22-0+5+LENGTH+2+8+2+10+5+ENT=6+5+0+30+0
118	F1050 DSC 17.0.0
	E1060 DØRGQ-100
	F1065FAC DS +2492
2	F1070ID DSA
NOR	F1080LNKL DSA
ž	F1090LNKR DSA
STO	F1100CELL DSA
	F1110 DC 1,
-	F1120 DS 5
N	F1130ENT TFM TF+6,ID+4
FOR	F1140 AM TF+6,4,10
NO	F1150 AM ENT-1,5,10
HER	F1160 TF CF+11,1-ENT
cộn	F1170 BNF *+36,CF+11
	F1180CF CF CF+11
	F1190 TF CF+11,-CF-11
	F1200TF TF ,CF+11
	F1210 AM TF+6,1,10
	F1220 BNR ENT+12,-TF-6
	F1230 AM ENT-1,1,10
12	F2010 TFL FAC,-CELL
11	F2020 BNF SETID,-ID
10	F2030 B7 TEST2
9	F2040SETID MM -ID,5,10
8	F2050 BD *+32,99
7	F2060 CF FAC-1
6	F2070 B7 *+20
5	F2080 SF FAC-1
	F2090 BD *+32,98

m	F2100 CF FAC-2
-	F2110 B7 *+20
0	F2120 SF FAC-2
3	FZI30TESTZ BNF SETL,-LNKL
N	F213D B7 TEST3
-	E2150TEST2 PNE CETD-LINKD
-	E2155 B7 DONE
01	F2160SETR MA FAC+-LNKR
	F2170DØNE TFL -CELL,FAC
16.57	F2180 B7 1-ENT
	F2190LENGTHDC 2,1
	F2200 DEND
2	GIIGO IF CF+II+I-EMT
	CITAD AN TEALALD
	C1130 TEM TF+6+15+6
and a	G112351ADR DS
	GIIDDENE B CALL IRANGEER VECTOR
00	
	ATTROCTT DRY
	CIDACLARR DSA
IN IT	GIGBGLNKE DSA
	610701D 054
	61660 D6860-100
	01020 DSC 11*0*0
	01040 DVLC22+0+5+LENGTH+2+8+2+10+5+ENT+6+5+0+30+0
-	61030 DAC 6.SETIND.7-0
	EIG20 DL FreeLees-D
-	A D NUMBER OPPO
1	ANAME SETIND
64	#STERE RELEADABLE
	AVESEMBLE RELOCATABLE
	*LIST CARD
kin filma	6
1	
•	4 II EFIIND DEAM
	3

•	
	ZZ SETIND 0646
	ZZSPS
	*LIST CARD
	*ASSEMBLE RELØCATABLE
0	*STØRE RELØADABLE
	*NAME SETIND
	*ID NUMBER 0646
0	G1000**** SETIND 0646
	G1010Q DS ,*+101
	G1020 DC 6,987898,5-Q
0	G1030 DAC 6,SETIND,7-Q
92	G1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	G1050 DSC 17,0,0
0_	G1060 DØRGQ-100
	G1070ID DSA
W	G1080LNKL DSA
🗢 ĉ 🔄	G1090LNKR DSA
- ock	GIIOOCELL DSA
5	GIIIO DC 1.
0	GIIIZCALL BTM ,*+11
Ţ	GIII4INITASDS **-5
Man -	G1120 DS 5
2 H -	GIIZZENT B CALL,,,TRANSFER VECTØR
-H3	GIIZSSLADR DS ++5
- NRII	G1130 TFM TF+6,1D-4
9.8	G1140 AM (F+6,4,10
	G1100 AM ENT-1,5,10
-	
-	G1190CF CF CF+11
-	
	G1210 AM TE+6+1+10
12	G1220 BNR ENT+24-TE-6
0"-	G1230 AM ENT-1-1-10
10	G2010 MA #+35CELL
e	G2011 ME *+23=-CEL1
0 -	G2020 BTAM-SLADR 78 CALL TO INTERNAL UTILITY ROUTINE
7-	G2030 MA CELL. 99
6	G2040 BTM ++11 +FØRTRAN CALLING LINKAGE TØ EXTERNAL SUBDRØGRAM
0	G2050SETDIRDS **-5
4	
3	

	G2060 DSA -ID
	G2070 DSA -LNKL
	G2080 DSA -LNKR
	G2090CELL. DSA
19-01-0	G2180 B7 1-ENT
	G2190QQ DAC 6,SETDIR,
	G2200 DVLC,5,SETDIR
	G2202 DAC 6, INITAS,
	G2204 DVLC.5. INITAS
-	GZZZOLENGTHDS ,QQ-1
	GZZ30 DEND
-	
-	
-	HTTER D2 2
	HILLSEVEL BEN TEALT
	HITTO DE TR
	H10801CT 927
	HTOLODYLON DEV
-	HIDESTAC DE +2492
	NT000 D0800-T00
121	H1020 020112040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040 04040
	H1000 DAFCSS-0+2*FENOLH+S*B*S*F0*2*EW1-0+2+0*0*0*0
	H1030 DAC 6+51R1HD+7+0
	H1020 DC \$498789845-G
	HIG100 02 *x+101 101+x* 50 00101H
1.	WAME STRIND
	ANY SSEMBLE RELOCATABLE
	#1121 CV60
	TT STRIND DEAR

•	
	ZZ STRIND 0648
	ZZSPS
-	
	*LIST CARD
	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
	*NAME STRIND
-	*ID NUMBER 0648
	H10100 DC .*+101
	H1020 DC 6.987898.5-0
-	H1030 DAC 6.STRIND.7-0
	H1040 DVLC22-0.5-LENGTH-2-8-2-10-5-ENT-6-5-0-30-0
	H1050 DSC 17.0.0
<b>)</b>	H1060 DØRGQ-100
	H1065FAC DS +2492
2	H1070DATUM DSA
9 ²	H1080ICL DSA
CK	H1110 DC 1,*
\$10	H1112CALL BTM ,*+11
0	H1114INITASDS ,*-5
-	H1120 DS 5
MM	H1125ENT B CALL,,,TRANSFER VECTØR
	H1126SLADR DS ,*-5
40-2	H1130 TFM TF+6,DATUM-4
	H1140 AM TF+6.4.10
	H1150 AM ENT-1,5,10
	H1160 TF CF+11,1-ENT
	H1170 BNF *+36 • CF+11
ə	HIIBOCF CF CF+II
	H1190 TF CF+11,-CF-11
12	H1220 DNK ENT=1-1-10
F1	$H_{2010}$ MA $*+35-10$
10	H2011 ME *+231C
9	H2020 BTAM-SLADR 78 CALL TO INTERNAL LITULITY POLITINE
	H2030 TEL FAC -DATUM
7	$H_{2040}$ TEL =99.FAC
6	H2180 B7 1-ENT
5	H219000 DAC 6. INITAS.
4	
3	

H2192	DVLC,5, INITAS
H2194	DC 2,1
H2196LENGT	HDS ,QQ-1
H2200	DEND
	· 原来的 · 是不会是非常能考虑了
11120	AM FREATESTED T
11740	VR 12+2***10
11110	
11000	DC JINTHIRWIAN
11012006.0	DALE TRANSMISSION OF ALLEN
LIG STRUCKL	DE TERINITETETETIT DE DET
TTOYODGIML	NACY DELETERS VALUES IN TORE CATE SETS
I TODIANOTE	PATTA SALA MALTALE SALA TATA SALA SALA SALA SALA SALA SALA
TODDERAS	TARKA FAILA INDA LEVA ANDA
TIGPALYS	TOT ROBES
	102 +5141/15740
	D21 45970 10 10 10 10 10 10 10 10 10 10 10 10 10
TIBETEVC	Declaration of the second s
11090	058/98-100141110
11020	020 014040
17890	BAECESBO*9*0#0EBC1H*5*9*5*70*3*EB1-9*2*0*30*0
11020	DVC R4KGE1 *1+0
11050 BLK	DC: 9*8835881/2-0
110700.01	02 7++101
	KOE1++304334-31433
	SET SEVERE ELEMENT OF THE SECOND S
¥1151 CARD	

•	
1.1	
	ZZ KGET 0672
-T-	
	ZZSPS
	*LIST CARD
	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
	*NAME KGET
	*10 NUMBER 0672
	110100 DC ++101
	110100 DS \$*101 11020 DC 6+987898-5-0
	11020 DC 0,707090,0-Q
10	11040 DVL C22-0.5.1 ENGTH-2.8.2.10.5.ENT-6.5.0.30.0
	11050 DSC 17+0+0
	I1060 DØRGQ-100
and the second	11061FAC DS +2492
W	11063BETA DS ,2630
e e	I1064STZERØDS ,2771
CCK	11065FXZ DS ,3099
045	I1066**** =9, .=5, )=6, +=3, \$=7, *=8, -=2, /=11, .=13, (=10, ==1, *=14.
0	II067TABLE DVLC, 2,9, 4,, 2,5, 2,6, 10,, 2,3, 4,, 2,7, 2,8, 10,, 2,2, 2,11
T	I1068 DVLC, 2,, 2,13, 2,10, 16,, 2,1, 2,14
RM-	IIO69PØINT DSA BUF+174
	11070INDV DS 2++ 06 ØR 10
л-ы	11072 DGM
	11074NEXT DS \$PØINT-4; DTA ØR DCA
.8	11075MES DMESSAS EXIT CALLED BY KGET(E)
	11070BUF DAS 89
-	LIOTA DC 19.9DUFTITO
	LIORONCODE DEA
	11120 DS 5
	III30ENT TEM TE+6.INPT-4
12	I1140 AM TF+6,4,10
	I1150 AM ENT-1,5,10
9	I1160 TF CF+11,1-ENT
8	I1170 BNF *+36,CF+11
7	I1180CF CF CF+11
6	I1190 TF CF+11,-CF-11
5	I1200TF TF +CF+11
9 .	I1210 AM TF+6,1,10
3	
2	

-		
-	11220	DND ENTITY TO C
G	11220	DNR ENT-129-1F-6
-	11230	AM ENI-191910
0	11240	
3	11310	
N	11310	DNE DEC - INDT
-	11320	TP _DRINT.DVM_2
-	11340	
	11345PEC	
N	11350	AM FOINTS2910
	11390	TEM DOINT DUE
	11300	MA DEVICE-INDT
60	11400	A DEVIC-DEVIC
	11410	RD TYPE-DEVIC
1185	11420	TEM INDV-10-10
(	I1430	B7 *+20
	I1440TYPF	TEM INDV+06+10
	I 1 4 4 3 * * *	CLEAR BUFFER IMAGE IS 87 INPUT CHARACTERS (BUF BUF+172).
OR)	I1444***	1 BLANK (BUE+174) AND 1 RECORD MARK (BUE+176)
X	I 1445RERE	TR BUE-1-STZERØ-1
TOC	11450	TR BUE+88.STZERØ
60	I1460ALLBI	BD *+24 INDV-1
	11462	RCTY
P. S.	11475	GET NEXT
LOR	I1476	BD *+24, INDV-1
INN	I1477	BC4 RERE
11ER	I148C	BLXM*+12,174(A1)
- Court	I148E	CM BUF-2(A1), 10
	I148G	BNE NBLK
	I148J	BCXM*-24,-2(A1)
- 60 -	I148L	B7 ALLBL
	I148NNBLK	TDM BUF+2(A1)
	I148PRKM	DC 1, *, *
6	I148R	BNR *+24,BUF-2(A1)
	I 148T	TFM BUF-2(A1),22,10
	I148V	BCXM*-24,-2(A1)
10	I148W	BNC3NØRED
	I149A	RCTY
8	I114C	WATYBUE
- 60 7	I114JNØRED	CM -PØINT,22,10
6	I114L MARK	BNE NØRKM
5	I1490	BNF ØKGØ, RMSW
· ·	I1492	WATYMES
3	I1500	CALLEXIT

-	
	11900 CVITEXII
	115250KG0 SF RMSW
- The second sec	11530 B7 REG
	11540NØRKM TE FACPØINT
	11550 CM FAC.70.10
	11560 BL #+32
	11565**** NUMERIC = 12
1	11570 TEM BETA-12-10
	LISSO B7 DØNE
	11500 CM EAC+41+10
	LIGOSESEE ALDUARETIC - 0/
	LIGODANA ALFRADETIC - 04
- 67	I TAGENEN NETTUED - DEE TADLE LOOK HD
10	11082**** NETTHER ++ USE TABLE LØØK-UP
10	11700 BLX *+12+FAC(A1)
10	
	11/20 IF BETA,TABLE(AT)
RM	IZOTODONE CF FAC-I,,TO
1 C	IZOZODEVIC DS
CK CK	12030RMSW DS ++-2
L.S	IZO40 CF BETA-1
	I2050 MF BETA, RMSW
-	I2060 CF RMSW
MA	I2070ØUT TE -NCØDE,BETA
🥟 🛔	I2180 BSBB1-ENT A CHARTER AND BECCED HYPE THE FLORE
10-B	I2194LENGTHDC 1, 1
and and a	I2200 DEND
6.	
	\$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE D.P.K.
	THE DO LADEFORAIC
. 0	TREDIARD VSA DEATCYDEATC
	JISSON MOR NUL DEALCY-INDI
	11320 BUE HEBED*-DELML
10	IIIHAGEC VAN GOINLASAID
1	
	11290 14 EBGAULSKN-S
	11360 BVE BERT-TUBLAT
	ITROCT IL LVCLLXS

2		
~ —	ZZ LCNTR 0637	
	ZZFØR	
~		
~	*FANDK0810	
	*LDISKLCNTR 0637	L CNTOOLO
•	FUNCTION LENTR (LST)	LENTODIO
	RETURN	LCNT0020
	END	LCNT0040
. / Lenge		
•		
<u></u>		
· ·		
W		
Lo Co		
OCK		
5		
¥		
IFOR		
NO-2		
8		
	EVD	15140050
•		T ZIMONTH
12	CALL SCINED IN LINKA LEINAU	1 21WUD50
····	FUNCTION LEINAM (LEI)	
10	#FDIOKFELNVWOE28	
8	*EWUDK0830	
- · ·		
6	77FOR	
5	AT I STATA OADS	
4		
3		
• ²		

	77 1 57144 0/20	
	77E@P	
•	*FANDK0810 *LDISKLSTNAM0639	
	FUNCTIØN LSTNAM (LST) LSTNAM=LNKL (CØNT (LST+1)) CALL SETDIR (O+LSTNAM+LSTNAM)	LSTN0010 LSTN0020
•	RETURN END	LSTN0040 LSTN0050
TOCK FORM		
R-UNIFORM		
соние	SDERF DEFANS DATE BARAGE FREDERAM FULC DAPAKA	
12	END BELOWN CONTRACTORY (CONT (CONT) )	ГСИ10040 ГСИ10030 ГСИ10030
11 10	*LDISKLCNIR 0637 FUNCTION LCNIR (LST)	LCNIGDIO
8 7	DIGONDROGIC BOJIC	
3	IX FCMIR ORAL	
3 _ 3 _		

1.00		
• <u> </u>	ZZ INTLBL 0649	
1	ZZEOR	
	*FANDK0810	
78.0	*LDISKINTLBL0649	
	FUNCTION INTLEL (I)	INTLOOIO
1.		INTLO020
	END	INTLO040
19 21		history and here and he
io		
W		
FOI		NUCLIARD
Lock		
бя		A CONTRACTOR OF A LOCAL AND A CONTRACT OF A LOCAL AND
		A STATE OF A
W		
NIFOR		NUCCOLPD.
- ni		
A Drank		
-		
	EWD	FLLL0040
	RETURN	FLLOGBO
11	LEAFER X	
10	ENVELLEN ELLER (X)	EFILOUID
9		
8	APANDYADIA.	
7	TTL SH	
5		
4		
3		
2		

	ZZ FLTLBL 0651	
	ZZFØR	
	*FANDKOBIO	
	*LDISKFLTLBL0651	
	FUNCTION FLTLBL (X)	FLTL0010
	FLTLBL=X	FLTL0020
	RETURN	FLTL0030
A	END	FLTL0040
1		
-118		
۰ 😸		
DRM		
27 E		
TOC		
Ŧ		
MA		
NIE 🧑		
ER-U		
our si		
2	SELOKU	INTEGO30
12		10110020
	FUNGTION (NTCOL LT)	
9	#EDISKINILSLO649	
8	*FANDKOBIO	
7		
6	5	
5	27 INTINI 0649	
4		

a la caracteria		
ത	ZZ NUCELL 0653	
	77F0R	
0		
(7)	*FANDK0810	
N	*LDISKNUCELL0653	
-	FUNCTION NUCELL (X)	NUCE0010
	K=1	NUCE0020
	10  M=LNKR(CONT(-1))	NUCE0030
	IF(M)2,4,2	NUCE0040
	4 GØ TØ (6,8),K	NUCE0050
	6 K=2	NUCE0060
•	CALL INITAS (X)	NUCE0070
- in	GØ TØ 10	NUCE0080
	8 TYPE 1	NUCE0090
	1 FØRMAT(18HNØ MØRE FREE SPACE)	NUCEDIDO
1 1 1 1 <u>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 </u>		NUCEO110
N N N N N N N N N N N N N N N N N N N	$\frac{2}{16} = \frac{1}{16} $	NUCEO120
¥ 4	$\frac{12}{12}  (ALL SETIND (-1,-1,-1,-1,NKR(CONT(M)),-1))$	NUCEO140
	CALL STRIND (0.M)	NUCE0150
0	CALL STRIND (0.M+1)	NUCE0160
	NUCELL=M	NUCE0170
	RETURN	NUCE0180
en rou	END	NUCE0190
NO.		
RIER		
60 R		
		REPEGOPO
		BCET DUGD
and the second second		8010040
		NCET 0030
12		SCLF0050
60 ¹¹	PORKEDITUR KCENT (CETT)	RCELOOIO
10	*FOISKK/EFF 0835	
9	*EVVDKOBIO	
8		
6		
	YY REELL DASS	
3		
2		
•		

112.2		
	77 RCELL 0655	
	ZZFØR	
	NEANDUARSA	
	*FANDK0810	
	*LUISKRULL 0655	D.C.F. LAND
	SUBROUTINE RCELL (CELL)	RCELOOIO
-	CALL SETIND (=1,=1,CELL,=1,=1)	RCEL0020
	CALL SETIND (-1,-1,-1)	RCELOOSO
	RETURN	RCEL0040
	END	RCEL0050
		North 0000
	EIGHTEND	BIICEO DO
	DE1/10W	MILLEVIUS
	ATTE STRATE CONNERS	
	TE CHEF DELEMO ILTERTICATIONICATION	ИПГЕОТНО
-	CWFF EXTL	HOCEOTTO
	8 IABE I	MICE0080
	Co 19 10	RUCEODAD
	CALL INITAS LAI	NUCEO070
	4 CO TO 16.01.K	
-	IC H=FNKH(CSWII-I))	MUCE 0030
-	HEDERKUSSEEFFDERS	
_	342.99	
	SS MACTER DESK	
-		

LUCAL DAY		
•		
A CARE		
-		
	ZZ MILISI 0657	
<u></u>	ZZFØR	
~	*FANDK0810	
	*LDISKMTLIST0657	
	FUNCTION MTLIST (LST)	MTLI0010
	L=LØCT(LST)	MTLI0020
N	IF(LISTMT(L)) 2,4,2	MTLI0030
	2 LR=LNKR(CØNT(L))	MTLI0040
	LL=LNKL(CØNT(L))	MTLI0050
	CALL SETIND (-1,L,L)	MTLI0060
	CALL SETIND(-1,-1,)LR,LNKL(CØNT(-1)))	MTLI0070
in	(ALL SETIND(-1)(L) - 1)	MTLIOOBO
6	4 MTLIST=!	MTL 10100
	RETURN	MTLIOIDO MTLIOIDO
	END	MTLIOI20
() (i) (i) (i) (i) (i) (i) (i) (i) (i) (	THE REPORT OF THE PARTY OF THE	MILIOILO
X		
510		
•		LDGROOSS IN INC.
RM		
€ <u></u>		
n-2	A RETURN (A)	1 BYEOTOO
		T DAUG 1 AD
8		TUNESTAN
1.1.1		A DELOTAR
	NEF BOUWWITT	T MACODAG
	V CHIE WILLSLIF	1887-0000
	TETIBALSTIZ ANA	TBWF0020
	IRALST*LCNTR(L)	1 RALOD40
12	CALL SETING(-1+-1+LCNTR(L)-1+L+L)	IRAL0030
11	F=Faci(Fei)	18410020
10		IRALODIO
9	*[0]81161164[310659	
B	ARVADAGE C	
7		
6	3-2-E-1-P	
5	15 LEVIEL DESC	
4		
3		
e 2		
	ZZ IRALST 0659	
---------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------
A		
	ZZFØR	
1	*FANDK0810	
	*LDISKIRALST0659	
	FUNCTION IRALST (LST)	IRALOOIO
-	L=LØCT(LST)	IRAL0020
	CALL SETIND(-1,-1,LCNTR(L)-1,L+1)	IRAL0030
	IRALST=LCNTR(L)	IRAL0040
	IF(IRALST)2,4,2	IRAL0050
	4 CALL MTLIST(L)	IRAL0060
	N=LSTNAM(L)	IRAL0070
19	IF(N)6,8,6	IRALOOBO
	6 CALL SETIND(1,-1,-1,L)	IRALO100
	CALL STRIND (N.L+1)	IRALO110
	8 CALL RCELL(L)	IRAL0130
	Z RETURN	IRAL0140
	END	IRAL0150
00		
5		
Γ		
ORM		
		WIFEASEA
1- 1-	LEVIA CONTRACTOR OF CONTRA	MALIVIAN GEFERTE
201KI		STREATES
	AUTE STITUSINESS TRAFTICATES	
		H1110000
-		WICTORIO
-	WET OFILID ISTATATATAT	WIFI0000
-		WIFT0020
	S CHEFTARK (CONTINUE)	WITIGORD
		W110030
12		WITTORSO
	ANMELING WIFTER (FRI)	
10		
9		
8		
7		
6		
· · · · · · · · · · · · · · · · · · ·	51 WITTEL 0003	
4		
3		
A 2		
and the state of		

1.50		
	ZZ LDUMP 0661	
	ZZEØR	
	*FANDK0810	
-	*IDISKIDUMP 0661	
	FUNCTION IDUMP (N1.N2)	I DUMOQ1Q
	FOULVALENCE (XW.IW)	
-	IT = 2 - N1 + N1/2 * 2	1 DUM0012
	C IT. 1=0DD POSITIVE. 2=EVEN. 3=0DD NEGATIVE.	LDUM0014
	LL = LNKL(CONT(-1))	LDUM0016
-	LR = LNKR(CONT(-1))	LDUM0018
	PUNCH 3, IT, N1, N2, LL, LR	LDUM0020
10	3 FØRMAT(11,5X,19HLDUMP ØF SLIP-CELLS,16,7H THRU,16,	LDUM0030
	1 10H AVSL=,216)	LDUM0031
10	LDUMP=0	LDUM0040
	GØ TØ (4,6,6),IT	LDUM0042
	6 PUNCH 5	LDUM0044
	5 FØRMAT(65HO ADDRESS ØF FIRST CELL MUST BE ØDD AND PØSITIVE-	-LDUMLDUM0046
5	1P ØMITTED)	LDUM0047
2	RETURN	LDUM0050
2.16.10	4 DØ 2 I=N1,N2,2	LDUM0060
	LL=LNKL(CØNT(I))	LDUM0080
2	LR=LNKR(CØNT(I))	LDUM0090
	IX = ID(CONT(I))	LDUM0100
	XW = CONT(I+1)	LDUM0120
S S S S S S S S S S S S S S S S S S S	LDUMP=LDUMP+1	LDUM0122
COU	2 PUNCH 1, I, LL, LR, IX, IW, XW	LDUM0130
	1 FØRMAT (1X, I5, 3X, 5HLNKL=, I5, 3X, 5HLNKR=, I5, 3X, 3HID=, I2, 3X,	LDUM0140
	1 01 0 6 HDATUM=, 111, 2X, 2HØR, E16.10)	LDUM0150
	RETURN	LDUM0160
11.	END T = 0	LDUM0200
		TOLADDAO
2	4 TELENKREKI-ENKRILIIZ#F#2	I GIVE D Q 3 O
1		100 EUG20
0		
	#LDTSKIOUAL DAAS	
8	*** **********************************	
7		
6	XXE SE	
5		
4		

27         IQUAL 0665           ZZF0R         **FANDK0910           **FANDK0910         IQUA0020           **FILKL(K)-LNKR(L))2,24.2         IQUA0020           4         IFLINKR(L))2,24.2         IQUA0020           6         IFLINKR(L))2,26.2         IQUA0020           8         IQUA020         IQUA0020           8         IQUA020         IQUA0020           8         IQUA1:0         IQUA0020           8         IQUA1:0         IQUA0020           8         IQUA1:0         IQUA0020           8         IQUA020         IQUA0020           9         IQUA020         IQUA0020           10UA0050         IQUA0050         IQUA0050           2         IQUA1:0         IQUA0050           8         IQUA020         IQUA0050           9         IQUA0050         IQUA0050           9         IQUA0050         IQUA0050           9         IQUA0050         IQUA0050           9			
22         IQUAL 0665           22F0R         *FANDKOBIO           *FANDKOBIO         IQUA0010           *LDISKIQUAL 0665         IQUA0020           FUNCTION IQUAL (K+L)         IQUA0020           4         IFLINKR(K)-LNKR(L))2.4x2         IQUA0020           6         IFLINKR(K)-LNKR(L))2.4x2         IQUA0020           6         IFLINKR(K)-LNKR(L))2.4x2         IQUA0020           6         IQUA0250         IQUA0050           7         IQUA0250         IQUA0050           8         IQUA0250         IQUA0050           8         IQUA0250         IQUA0050           2         IQUA0250         IQUA0050           2         IQUA0250         IQUA0050           2         IQUA0250         IQUA0050           2         IQUA0250         IQUA0050           3         IQUA0250         IQUA0050           4         IQUA0250         IQUA0050           5         IQUA0250         IQUA0050           6         IQUA0250         IQUA0050           6         IQUA0250         IQUA0050           6         IQUA0250         IQUA0050           6         IQUA0250         IQUA0050			
22F6R  *FANDKOB10 *LDTSKIQUAL 0665 FUNCTION IQUAL (K+L) IQUA0020 4 IF(LNKL(K)-LNKL(L))2;4+2 IQUA0020 6 IF(ID(K)-ID(L))2;8+2 IQUA0030 6 IF(ID(K)-ID(L))2;8+2 IQUA0050 8 IQUAL=K-L IQUA0050 8 IQUAL=K-L IQUA0050 2 IQUA0050 8 IQUAL=K-L IQUA0050 8 IQUA00		ZZ IQUAL 0665	
*FANDKOBIO           *LDISKICUAL 0665           IF(LNKCK)-LNKL(L))2,4+2           IGUA0020           4         IF(LNKCK)-LNKR(L))2,4+2           IGUA0020           6         IF(LNKR(K)-LNKR(L))2,4+2           IGUA0020           8         IGUA0400           8         IGUA0400           8         IGUA050           8         IGUA0600           8         IGUA010           9         IGUA0600           8         IGUA010           9         IGUA020           8         IGUA020           8         IGUA020           8         IGUA020           9         IGUA020		ZZFØR	
#FANDKOBIO           #LDISKIOUAL 0665           FUNCTION IGUAL (K,L)           IF (LNKR(K)=LNKR(L))2,4+2           IGUA0020           4           IF (LNKR(K)=LNKR(L))2,4+2           IGUA0020           6           IF (LNKR(K)=LNKR(L))2,4+2           IGUA0020           6           IF (LNKR(K)=LNKR(L))2,4+2           IGUA0020           6           IF (LNKR(K)=LNKR(L))2,4+2           IGUA0050           6           IGUA050           8           IGUA050           9           9           9           9           9           9           9           9           9           9           9           9 <t< td=""><td></td><td></td><td></td></t<>			
*LDISKIQUAL 0665         IQUA0020           IF(LNKLKK)-LNKL(L))2,*+2         IQUA0020           4         IF(LNKRKL))2,*+2         IQUA0020           4         IF(LNKRKL))2,*+2         IQUA0020           6         IF(LDKRKL))2,*+2         IQUA0020           6         IF(LDKRKL))2,*+2         IQUA0020           6         IF(LDKRKL))2,*+2         IQUA0020           6         IF(LDKRKL))2,*+2         IQUA0020           7         IQUA0250         IQUA0050           8         IQUA0250         IQUA0050           8         IQUA0050         IQUA0050           2         IQUA0050         IQUA0050           8         IQUA0050         IQUA0050           9		*FANDK0810	
FORCTION IGUAL (K):         IGUA0010           IF(LNKL(K)-LNKL(L))2:6:2         IGUA0020           6         IGUA002         IGUA0020           6         IGUA002         IGUA0020           6         IGUA002         IGUA0020           6         IGUA0020         IGUA0050           7         IGUA0020         IGUA0050           8         IGUA1=K-L         IGUA0050           8         IGUA1=0         IGUA0050           8         IGUA1=0         IGUA0050           8         IGUA01=0         IGUA0050           9         IGUA01=0         IGUA0050           8         IGUA1=K-L         IGUA0050           9         IGUA1=K-L         IGUA0050           9         IGUA010         IGUA0050           9         IGUA010         IGUA0050           9         IGUA0050         IGUA0050		*LDISKIQUAL 0665	
IF(LNRE(K)-LNRE(L))2:9:2         IQUA0020           6         IF(LNRE(K)-LNRE(L))2:9:2         IQUA0020           6         IF(L0K)-ID(L)2:8:2         IQUA0020           8         IQUA1=X-L         IQUA0020           8         IQUA010         IQUA0020           8         IQUA0020         IQUA0020           8		FUNCTION IQUAL (K,L)	IQUA0010
*         IFULNIKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKK/FUNKKFUNKK		IF(LNKL(K)-LNKL(L))29492	1QUA0020
0         I GUAL=K-L         I GUAOSO           8         I GUAL=C         I GUAOSO           9         I GUAL=C         I GUAOSO           2         I GUAL=C         I GUAOSO           6         E NO         I GUAOSO           8         I GUAOSO         I GUAOSO           9         I GUAL=C         I GUAOSO           9         I GUACESO         I GUAOSO           9         I GUAOSO         I GUAOSO           9         I GUAOSO         I GUAOSO           9         I GUACESO         I GUAOSO           9         I GUACESO         I GUAOSO           9         I GUAOSO         I GUAOSO           9         I GUACESO         I GUAOSO           9         I GUAOSO		4 IF(LNKK(K)=LNKK(L))29092 6 IE(ID(K)=ID(L))2.9-2	I QUADOSO
CO         IGUAL=0         IGUA0050           RETURN         IGUA0050           2         IGUAL=1           RETURN         IGUA0050           RETURN<			10040040
A     IQUA0050       2     IQUA11       RETURN     IQUA0050       RETURN     IQUA0080       END     IQUA0090       RETURN     IQUA0090	-		IQUADOSO
2     IQUAL=1     IQUA0070       RETURN     IQUA0080       END     IQUA0090       RETURN     I		RETURN	IQUADDED
RETURN I OUADOBO IOUADOBO IOUA		2 IQUAL=1	IQUA0070
S         E ND         I QUA0090           I QUA0090         I QUA0090           I MACCMULTI         I QUA0090           I QUA0090         I QUA0090           I QU		RETURN (TRATE ART AND AND A TRATE AND A TRADE AND A	IQUA0080
22         LDUMP 0461           23         LDUMP 0461           24         DDUMP 0461           25         LDUMP 0461           25         LDUMP 0461           26         LDUMP 0461           27         LDUMP 0461           26         LDUMP 0461           27         LDUMP 0461           28         LDUMP 0461           29         LDUMP 0461           20         LDUMP 0461           21         LDUMP 0461           22         LDUMP 0461           23         LDUMP 0461           24         DDUMP 01           25         LDUMP 0461           26         LDUMP 0464           27		S ENDCHOTATEDC+DH+LX+TM+XM	IQUA0090
27         LDUMP 0661           22F0R         #FANDK0810           22F0R         #FONKC1200           #FONKC1200         EQUIVALENCE 1X%.191           #FONKC1200         EQUIVALENCE 1X%.182           #FONKC12000         EQUIVALENCE 1X%.14.68           #FONKC120000         EQUIVALENCE 1X%.182           #FONKC1200000         EQUIVALE			
ZZE RR           ZZE RR           ZZE RR           RFAMDK0810           FUNCTIONE 0641           FUNCTIONE 06411           FUNCTIONE			
ZZ LDUMB OK61           ZZERR           ZZERR           ZZERR           #FANDKOBIO           #FONKCIEN           #LOUMP OK61			C0040106
22         LBUMP 0661           22         LBUMP 0661           22         ZZERR           23         ZZERR           24         ZZERR           25         ZZERR           26         ZZERR           27         ZZERR           28         ZZERR           29         ZZERR           20         ZZERR <td></td> <td>LR*LNKRICONT(I)</td> <td>TONWOORD</td>		LR*LNKRICONT(I)	TONWOORD
ZZ         LDUMP         D661           ZZF8R         #FANDK0810           #FOISKLDUMP 0661         EDUMP012           #LDUMP 0661         EDUMP012           #LDUMP 0661         EDUMD010           EQUIVALENCE (XW.41%1         EDUMD010           EQUIVALENCE (XW.41%1         EDUMD010           EQUIVALENCE (XW.41%1         EDUMD010           EQUIVALENCE (XW.41%1         EDUMD010           IT=2+NYLICENTIVE, 2+FVEN% 3+#DD NEGATIVE, 0UND010         EDUM0010           IT=2+NYLICENTIVE, 2+FVEN% 3+#DOWP 0F SECULES, 0UND010         EDUM0010           IT=2+NYLICENTIVE, 0UND0         EDUM0010		TETUKI (CSM1(T))	I DOMOGEG
ZZ         LDUMP 0661           ZZF08R         #FANDK0810           #FONKCI10N LOUNP (N1.A23)         LDUM0010           #LDISKLOUNP 0661         EDUM0010           #LUNK1(EBN/1/2)*         Servens 3+000 MeGATIVE*           #LDIM001         EDUM0010           #LUNK1(EBN/1/2)*         EDUM0010           #LDIM000         EDUM0010           #LOUNP 0660         EDUM000           #LOUNP 0660         EDUM000           #LOUND 0760         EDUM000           #LOUND 0760         EDUM000           #LOUND 0760         EDUM000           #LDUM0000 </td <td></td> <td>The Strangers</td> <td>T DIMOUVU</td>		The Strangers	T DIMOUVU
ZZ         LDUMP         DAMI           ZZF 88         225 88           FEANDK 0810         ECUNCTIENC INN.100           #LDISKLDUNP 0641         ECUNCTIENC INN.100           FEUNCLENCE INN.100         ECUNCTIENC INN.100           INN.10000000000         ECUNCTIENC INN.100           INN.1000000000000000000000         ECUNCODE           INN.100000000000000000000000000000000000			Though and
ZZ         LBUMP 0661           ZZFERR         ZZFERR           ZZFERR         EDWR010           FINCTION LOUND INT.N21         EDWR010           FINCTION LOUND INT.N21         EDWR010           FINCTION LOUND FINT.N21         EDWR010           FINCTION LOUND FINT.N22         EDWR010           FINCTION LOUND FINT.N230         EDWR010           FINCTION LOUND FINT.N230         EDWR010           FINCTION FINT.N230         EDWR010		ID WAILLEVI	
ZZ         LEUMP 0661           ZZERR         ZZERR           ZZERR         EDWR20661           FONCTION LOUNP (N1.02)         EDW0010           EUNALENCE (XK.18)         EDW0011           EUNALENCE (XK.18)         EDW0012           EUNALENCE (XK.18)         EDW0012           EUNALENCE (XK.18)         EDW0012           EUNALENCE (XK.18)         EDW0012           EUNALENCE (XK.18)         EDW0016		E FRENKTIKKNO INTELECTION FIRST FEIT MUST BE MAR IND BASTTIVEL	
ZZ         150MP_OK61           ZZFØR         ZZFØR           ZZFØR         LDUMPOLA           #FANDKOSIO         LDUMODIO           #LDUMP_OK61         LDUMODIO           #LDUMODIO         LDUMODIO           LDUMODIO         LDUMODIO           IT=2=M14MI/252         LDUMODIO           C         IT=2=M14MI/252           C         IT=2=M14MI/252           LDUMODIE         LDUMODIE           LINKLICENT         LDUMODIE			Chowhows
IDUMP_ORGI           ZZEGR           ZZEGR           FEANDKOBIO           #LDISKLDUMP_OLEI           FEANDKOBIO           FEANCTIEN           FEANDKOBIO           FEANDKOBIO           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANCTIEN           FEANTALIENCE IXM.INI           FEANTALIENCE IXM.INI           FEANTALIENCE IXM.INI           FEANTALIENCE IXM.INI           FEANTALIENCE           FEA			CDAW0040
22         LBUMP 0661           22         LBUMP 0661           22         STERK           23         Sterk           24         Sterk           25         Sterk           26         Sterk           27         Sterk           28         Sterk           29         Sterk           29         Sterk           21         Sterk           21         Sterk           21         Sterk	1000		
22         LDUMP 0661           22FERR           *FANDK0810           *LDISKLOUMP 0661           FUNCTION           FUNCTION           ECUNCTION           ECUNCTION </td <td></td> <td>1 FRENVISTIANTONNE GE STIB-CEFF241041H INEG4104</td> <td>FDIW0030</td>		1 FRENVISTIANTONNE GE STIB-CEFF241041H INEG4104	FDIW0030
ZZ LDUNP OK61 ZZF8R PFANDK0810 *LDISKLDUMP 0661 FUNCTIRN LDUMP (N1.N2) EQUIVALENCE (NW.19) LDUMD010 IT=2=N1+N1/2%2 LDUMD014 LL=LNKL(CRMII=L1) LDUM0016 LDUM0016 LDUM0016 LDUM0016 LDUM0016 LDUM0016		BANCH DILLINGING IN THE	T BAWODNO.
ZZ LDUMP OK61 ZZFOR ZZFOR #LDISKLOURP OK61 #LDISKLOURP OK61 EUNCIION LOUMP (NI.423 EQUIVALENCE (XW.1W1 EQUIVALENCE (XW.1W1) EQUIVALENCE (XW.1W1 EQUIVALENCE (XW.1W1) EQUIVALENCE (XW.1W1 EQUIVALENCE (XW.1W1) EQUIVALENCE (XW.1W1) EQUIV			
IZ LOUNE OKGI ZZFERK PFANDROGIO PLOLUNE OBEI PLOLISKLOUNE OBEI FUNCTIEN LOUNE (NI.K2) EGUIVALENCE IXW.INI II.2.MI4NI7282 C II I-800 FECITIVE. 246VEN. 34000 NEGATIVE. LOUNDOIN		FT=FMKT(CSW1(-T))	T DOMOD16
ZZ LOUMP OS61 ZZFOR ZZFOR #LDISKLOUMP OS61 #LDISKLOUMP OS61 FUNCTION LOUMP (MI.M23 FOUNCION LOUPP (MI.M23 FOUNCION			
ZZ LOUMP OK61 ZZFOR PFANDKCGIO #LDISKLOUMP OK61 FUNCIISN LDUMP (NI.M2) ECUIVALENCE (XW.1W) LDUMOGID			
ZZ LBUMP DA61 ZZEOR ZZEOR FFANDROEIO FLONCIEN LOUMP (KI.442)			
ZZ LEUMP OKAJ ZZFER FFANDKOGIO FLOUND CHAI			
ZZERR FEAMORCEIG	_	ALDISKLOUMS OBEL	
ZZ LOUND OGAI			
22 FOUND 0261		A C S F BR	
	-		

•		
	77 LICTMT 0447	
N	ZZ LISIMI 0007	
0	ZZFØR	
~		
N	*FANDK0810	
-	*LDISKLISTMT0667	1.1070010
	I = I ØCT (I ST)	
~	LISTMT=IQUAL(CØNT(L) • CØNT(LNKR(CØNT(L))))	LISTOOR
6	RETURN	LIST0040
	END	LIST0050
185		
••• *	THE REPORT OF THE LEW STRUCTURE PERSON FILE ST D. P. C.	
Ma		
0		
OCK		
5		
W		
E CO		
AU-R		
DURIE		
	E MO	NAM10050
	2 NAMISTAL	
1	651060	
	A MANTET-INULIFERENTI NEDICEPTI NELICENTI JIII.CONTI IL	
12		NAMIOOXO
	FUNCTION NAMIST (L)	
	*LDISKNAMISTOCCS	
6	*E YVDKOBIO	
6	6 <u> </u>	
600	2 VI NAMISI CESS	
4		
2		
•		
Ing		

-	ZZ NAMTST 0669	
	ZZFØR	
-		
	*FANDK0810	
-	*LDISKNAMTSTO669	NAMTOOLO
	$\frac{1}{1} = 1 \times 1$	NAMT0020
	4 IF(ID(CØNT(L))-2)2,6,2	NAMT0030
	6 NAMTST=IQUAL(CØNT(LNKR(CØNT(LNKL(CØNT(L)))), CØNT(L))	NAMT0040
	RETURN	NAMT0060
	2 NAMTST=1	NAMTOO 70
	E I URN E ND	NAMT0080
1185		
RM		
	×	
100		
NIN N		
1		
		F1910090
	LISTMT-IOUAL(CONT(L),CONT(LNKR(CONT(L)))	
1		
. 🧶 ,	EUNCTION LISTAT (CST)	
	9 61 01 Ch1 1 C AN 4 0 C C 1	
1	8	
	差之 <b>千</b> 億秋	
	5	
	4 IS LIKIMI OKAT	
	3	
	2	

٠		
1.1.1.1.1.1		
~	ZZ LØCT 0671	
N	THE REPORT OF TH	
0	ZZFØR	
3	*EANDRORLO	
2	*FANDRUGIU	
	FUNCTION LACT (LST)	LACTOOLO
	IF(NAMIST(IST))2.4.2	10010
~	4 10CT=1 ST	10010020
	RETURN	LØCT0040
	2 TYPE 1.LST	LØCT0050
	1 FØRMAT(16HØPERAND NØT LIST,111)	LØCT0060
	CALL EXIT	LØCT0070
8	END BELORE CLASS	LØCTOO80
•	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE D.P.K.	
A CONTRACTOR	WW 12140+0*70	
X	▲ 新聞公式: 法犯罪 美型保留美国公司发展一步	
🐑 Ö		
ock	DC I*;	
sto		
•	KSED DAY	
	ILHVS DAV	
ORM		
4 NO		
<u> </u>	2642 26 26 26 26 26 26 26 26 26 26 26 26 26	
ano	08823-100	
	020 13*0*0	
	DATC35-0+2+TENO1H+5+8+5+10+2+EN1-0+2+0+80+0	
	CONTRACTOR OF A	
•	Research Koni Dest	
12	*ID NUMBER ORDA	
11	WVWE KERI	
10	KASSENBLE RELOCATABLE	
9		
6	RELOOD DELOTOTEL	
7	TTOLS	
6		
5	13 KPUT 069%	
4		
3		

	ZZ KPUT 0694
	ZZSPS
	*STØRE RELØADABLE
	*LIST CARD
	*ASSEMBLE RELØCATABLE
	*NAME KPUT
	*ID NUMBER 0694
	**** KPUT 0694
	Q DS ,*+101
	DC 6,987898,5-Q
	DAC 6, KPUT ,7-Q
10	DVLC22-Q.5.LENGTH.2.8.2.10.5.ENT-6.5.0.30.0
81	DSC 17.0.0
· · · · · · · · · · · · · · · · · · ·	DØR60-100
	FAC DS 2492
	FX7 DS • 3099
OR	BETA DS +2630
×	IPNT DS 5
100	ICHAR DSA
· · · · · · · · · · · · · · · · · · ·	WORD DSA
	IST DSA
[	DC 1at
ORM ORM	
- INIE	ENT TEM TELS ICHAR-A
	AM TE+6+4-10
	AM ENT 1 5 10
	PDEGE DEL AM ENTEISOSIU DESERVI ILE - D'O'K
1.1.1	IF (F+II)=ENI
-	BNF *+36,CF+11
•	CF I CF I CF I I
	GETCCE+11++CF+11
Line States	TEADE TF (+CF+11
	AM TF+6,1,10
12	BNR ENT+12,-TF-6
	AM ENT-1,2,10
10	BSBA*+12
9	CM -WØRD,,10
8	BNE *+24
7	TFM IPNT
6	AM IPNT,2,10
5	CM IPNT, 12, 10
	BNE NØT

-	
1	
4.	BTM 9*+11 NEWPOTDS -*-5
	DSA -WØRD
	DSA -I ST
	TF -WØRD+FXZ
	TFM IPNT,2,10
•	NØT BLX *+12,IPNT(A1)
1.1.2.1.4.1940	TF FAC,-WØRD
N	TF BETA,-ICHAR
•	TD FAC-11(A1),BETA-1
1.17	TD FAC=IU(AI),BEIA
-	TE -WORD FAC
	BSBB1-ENT
	QQ DAC 6, NEWBØT,
•	DVLC +5 + NEWBØT
	DC 2,1
W	LENGTHDS ,QQ-1
€ £	DEND
õ	SDERE DEETNE END RE S. P.S. PRRCRAM ETLE R.L.S.
in	The print and of the the tree the tree the tree the tree the tree the tree tre
<u> </u>	
e la	
NU-F	
RIEF	
S	
1. The second second	
-	
	EVER SEIDIELO-FISI-FISI-FISI-
12	
0	EUNCLISM FISIKI
9	ALDISKUIST 0675
8	AFFANDED TO
6	22620
5	
4	
3	
2	

-	77 LIST 0675
4	22 2151 0875
	ZZFØR
	*FANDK0810
	*LDISKLIST 0675
	FUNCTION LIST(K)
	LIST=NUCELL(Z)
	CALL SETDIR(0,LIST,LIST,LIST)
-	CALL SETIND(2)LIST(LIST)
	1F(1QUAL(K)9)/20102
100	Z CALL SETIND(=19=1919LISI+1)
	1 RETURN
1.1.8	END
3	
×	
TO	
	20203-021-021-192 290 01 2*0*2* 6650698 11CF - x*f*2*
W	
NU-P	REMOTHDEN AQORT CHURCH
Ca la	
6. 6	DALCAS ANEWERSTO
	CO DVC C*WEREREATT
	The second s
13	
1	
1	
-	
-	

•	
LC?	ZZ TØP 0677
2	ZZFØR
~	NEANDUARDA
2	*FANDK0810 *IDISKTØP 0677
•	FUNCTION TOP(P)
	TØP=CØNT(LNKR(CØNT(LØCT(P)))+1)
N.	END
•	
-	
STO	
e a	
010	
•	
•	
•	EVP
3	
	EUNCTION BOILDI
	*FDISK961 0018
	STERE
-	5 EWI 07438
1	

- Inde	
	ZZ BØT 0679
	ZZFØR
-	
	*FANDK0810
	ALDICERAT 0470
-	*LUISKOWI UCIA
	FUNCTION BOT(P)
	BØT=CØNT(LNKL(CØNT(LØCT(P)))+1)
	RETURN
	END
100	
50	
3	
No B	
¥	
00	
5	
-	
- WI	
Eo.	
NO	
ER S	
- Ano	
. 0	
A 11	
10	
9	
8	
7	
6	
4	
3	
2	
-	
-	
0	

•	
d	ZZ PØPTØP 0681
N	ZZFØR
2	
2	*FANDK0810
60	FUNCTION POPTOP(P)
	PØPTØP=DELETE(LNKR(CØNT(LØCT(P))))
N	RETURN
	END
•	
1185	
•	
FORM	
OCK	
es) to	
NHO	
-UNIFG	
RIER	
Co 💿	
۲	
	END
12	RETURN
60 H	
10	#FDI2KBSbBSL0083
6	*FANDROBLO
	X S & SH
•	
•	77 DODRAT OGRA

-	
P	ZZ PØPBØT 0683
	ZZFØR
-	*FANDK0810
-	FUNCTION POPBOT(P)
	PØPBØT=DELETE(LNKL(CØNT(LØCT(P))))
	RETURN
	END
10	
6-118	
-	
- LOR	
CK	
sto	
[ —	
FOR	
IND-2	
URIEL	
······································	
	EVE RELOKE
12	6SB1GE=DEFE1E(FMXE(CB91(PECE(61)))
9	erb12kb0b1tbb0021
8	
7	
6	
з	
2	
-	

-	
	ZZ NEWTOP 0685
0	ZZFØR
3	*FANDK0810
2	*LDISKNEWTØP0685
•	FUNCTION NEWTOP(P;Q)
	NEWTØP=NXTRGT(P,LØCT(Q)) RETURN
No.	END
-	
•	
•	
•	
•	
•	
0	FUNCTION NEWBOILD (0)
	** DIVENERALOPEL
-	e · · · · · · · · · · · · · · · · · · ·
	7
•	4
	3
•	

1	
also and	
	ZZ NEWBØT 0687
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	ZZFØR
	*FANDKO810
State State	
-	*LDISKNEWBØIG087
	FUNCTION NEWBOI(P,Q)
	NEWBØT=NXTLFT(P,LØCT(Q))
Mg Para Bullion	RETURN
	END
0	
-	
· · · · · · · · · · · · · · · · · · ·	
-	
N N	
S 8	
ž	
0	
9	
T	
NN	
2	
No.	
ino uno	
-	
. 12	
H.	
10	
B	
- 7	
6	
5	
а	
2	
-	

00	ZZ NXTRGT 0689
2	775 00
0	ZZFØR
3	*EANDRORIO
2	*FANDRUOIU
1	*LDISKNXTRG10689
-	FUNCTION NXIRGI(M,A)
N	
	CALL SETIND(-1-1-1-D-A)
-	CALL SETIND(0-A-ID-ID)
-	IF(NAMIST(M1)1-2-1
1011	2 (ALL SETIND(1+=1+=1+IR)
(41)	CALL SETIND(-1) - 1 + 1 + 1 + 1 + 1 + 1
	1 CONTINUE
	CALL STRIND(M+IR+1)
	RETURN
	END
TO	
0	
0	
	Se Miler
1	
1	EALL SETTIO1-14-14LCNT91M(+14M41)
	S CVC 2014011+-1+-1+10
	CALL SETINDI-1.1L.+1.AI
6	
1	ZUAILE1#11
the second to	IT PRICE   131
1 1 W	EINTION MAIIFIN.EI
in the second	
1. 1. 1.	5 A A C U D
100	
1	

_	
97	ZZ NXTLFT 0691
	ZZFØR
•	
	*FANDK0810
	*LDISKNXTLFT0691
·	FUNCTION NXTLFT (M,A)
	IL=NUCELL(Z)
-	$\frac{11 - 1 \text{ NKL}(CONT(A))}{11 - 1 \text{ NKL}(CONT(A))}$
	CALL SETIND(-1,-1,IL,LL)
	CALL SETIND(-1,IL,-1,A)
	CALL SETIND(0,LL,A,IL)
ID.	IF(NAMTST(M))1,2,1
	2 CALL SETIND(1,-1,-1,IL)
	CALL SETIND(-1,-1,LCNTR(M)+1,M+1)
	1 CALL STRIND(M,IL+1)
	END
	END
202	
· · · · · · · · · · · · · · · · · · ·	
IFOR	RETURN
NO-te	CYEF LIBINDIN*(B+1)
9 ·6	
	S LYPE CELEVIELITE I COL
	CVTT-RELIVOL-TATES CONTRACTOR STATES CONTRACTOR STATES AND A STATES
	IR STARTING (CONTIAN)
11	
10	
9	el DISKNX3K0106AD
8	
7	
6	
5	
4	
3	

σ,	ZZ DELETE 0693
~ _	77 E 0 D
9	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	*FANDK0810
~	*LDISKDELETE0693
	FUNCTION DELETE(K)
	→ IF(ID(CØNT(K))-2)1,2,1
N	2 PRINT 100
	100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)
	DELETE=0. I (Most is
	RETURN
· · · ·	I DELETE=CØNT(K+1)
10	LL=LNKL(CØNT(K))
	CALL RCELL(K)
	CALL SETIND(-1+-1+1R+LL)
	CALL SETIND(-1+LL+-1+LR)
	RETURN
č.	END
510	52 CALLIMERARILWERDYTOP(SIACK))
•	
τ	C EVD B5"F#21
WH	
	THE REAL MENLES (SEERTIFICK))
	19716D=021
A second second	
12	
10	
9	ST SHONESIG
B	
7	
6	
0	1121
3	
2	

-	
1	ZZ LISTRD 0695
	ZZFØR
•	
	*FANDK0810
	*LDISKLISTRD0695
-	FUNCTION LISTRD(LST, IN)
	EQUIVALENCE (NUM, WØRD)
	CALL KGET(-1,ICØDE)
	ISUB=1
	10 ICHAR=KGET(IN,ICØDE)
	IF(ICØDE-10)10,20,10
•	20 LISTRD=LST
12	CALL NEWTØP(LST,LIST(STACK))
11-5	40 IDELSW=0
	NUM=0
- Section	IFLTSW=0
ž	ISINSW=0
	50 ICHAR=KGET(IN,ICØDE)
CK	IF (ICØDE) 51,51,54
STO	C END ØF LIST
	51 IF(IDELSW)53,53,52
	52 CALL NEWBØT(WØRD, TØP(STACK))
W	53 CALL IRALST(STACK)
💌 🔝	RETURN
4n-e	54 GØ TØ (110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230,
RIE	1 240),ICØDE
	C EQUAL SIGN
	110 IF(IDELSW)112,112,111
	111 CALL NEWBØT(WØRD, TØP(STACK))
٠	112 NUM=0
	CALL KPUT(ICHAR, WØRD, TØP(STACK))
	CALL NEWBØT (WØRD, TØP (STACK))
•	GØ TØ 40
12	C MINUS SIGN
11	120 IF(IDELSW)121,121,111
10	121 IDELSW=1
9	ISINSW=-1
8	CALL KPUT(ICHAR, WØRD, TØP(STACK))
	GØ TØ 50
6	C PLUS SIGN
5	130 IF (IDELSW)131,131,111
4	131 IDELSW=1
3	
2	

-	
The state	
-	ISINSW=1
<u> </u>	CALL KPUT (ICHAR + WORD + TOP (STACK))
(i)	GØ TØ 50
~	CLETTER
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	140 IDELSW=1
· · · · ·	CALL KPUT(ICHAR, WORD, TOP(STACK))
0	GØ TØ 50
	C DECIMAL PØINT
	150 IF(IDELSW)112,112,152
	152 IF(IFLTSW)153,111,111
	153 WØRD=NUM
1	SIGN=ISIGN
60	IFLTSW=1
in	DEC=10.
	GØ TØ 50
• • •	C RIGHT PAREN
	160 IF(IDELSW)162,162,161
X	161 CALL NEWBØT(WØRD,TØP(STACK))
e e	162 CALL PØPTØP(STACK)
č	I SUB=I SUB-1
ST0	IF(ISUB)163,163,40
	163 CALL IRALST(STACK)
	RETURN
W	C DØLLAR SIGN
1 🕖 Ê	170 GØ TØ 110
NO-2	CASTERISK
RIE	180 GØ TØ 110
CO CO	C BLANK BELLE
	190 IF(IDELSW)50,50,191
	191 CALL NEWBØT (WØRD, TØP (STACK))
1 ND	GØ TØ 40
	C LEFT PAREN
	200 IF(IDELSW)202,202,201
	201 CALL NEWBØT(WØRD, TØP(STACK))
12	202 ISUB=ISUB+1
	CALL NEWBØT(LIST(9), TØP(STACK))
10	CALL NEWTØP(BØT(TØP(STACK)),STACK)
	GØ TØ 40
8	C SLASH MELICAVE-JOINTER
- 💚 न	
6	C NUMBER 200
1 . s	220 IF(ISINSW)223,221,223
₩	221 1F(IDELSW)222,222,224
3	222 ISINSW=1
2	

tes .

•	
1	
	SAN ISINGREI
	223 ISIGN=ISINSW
-	ISINSW=0
	IDELSW=1
	IFLTSW=-1
_	NUM=(ICHAR-70)*ISIGN
	GØ TØ 50
	224 IF(IFLTSW)225,140,226
	225 NUM=10*NUM+(ICHAR-70)*ISIGN
	GØ TØ 50
-	226 VAL=ICHAR-70
and the second second	WØRD=WØRD+SIGN*VAL/DEC
	DEC=DEC*10.
()	GØ TØ 50
12	C CØMMA
-	230 GØ TØ 110
	C APPØSTRØPHE
	240 GØ TØ 140
X	END
A 5	
ž	
015	CONTRACTOR SELDENT CONTRACTOR STOLEN ST
5	
	TAR CALCURSPICE(STACK)
NO CINI	
8	TOOLEL FIDE FRATEONITOS FIDE THE FIDE STOLEN S
our our	C READING READER IN THE READER OF T
-	
12	368 FEITHER CUITING 169
10	
10	
9	ANTE ANDLA DENNE ANDLY INDICLYCKIN
8	
7	
6	
5	
2	

-	
	ZZFØR
0	
5	ZZ SEQRDR 0697
~	NE AND Y COLO
V	*FANDK0810
-	*LDISKSEQRDR0697
	SEORDR=CONT(LOCT(LST))
1 7	RETURN
0	END
0	
88	
OR	
X	
STO	
0	
T	
RM	
NIFO	
ER-U	
our	
0	
0	
	SETURN
	[-12101=N
0	SEOLR+CBMIIL+11
12	
11	
10	
9	
0	
	11100
5	
W 4	
3	
2	
-	

-

1.11.	
	ZZ SEQLR 0699
	77E @D
	ZZFØR
	*FANDK0810
21.00	*LDISKSEQLR 0699
	FUNCTION SEQLR(Z,N)
	Z = CONT(L)
	N=ID(Z)=1
	RETURN
	END,
-	
	SEGRER*CHILLESCI(USII)
	aTOIIX260KDK0201

	77 SEOLI 0701
33	
0	ZZFØR
S	*FANDK0810
	*LDISKSEQLL 0701
•	FUNCTIØN SEQLL(Z,N)
N	Z = CONT(L)
0	SEQLL=CØNT(L+1)
	RETURN
0	END
1185	
O °	
LORN	
OCK	
C) NBO	
linu-	26 16 3
OURIER	Z DALAKRIOSATICGATICATICATIC
	N=1D[]]+]
	ZYCRNT(L)
0	
	4 F=FUKB121
0	
13	IETID(1)=1)=====
<b>O</b> 10	FUNCTION SEOSE(2.N)
	RIDI2602010103
0	
6	
0	53 2E085 0303
3	
2	

•	ZZ SEQSR 0703
	ZZEØR
•	
	*FANDK0810
_	*LDISKSEQSR 0703 ELINCTION SEOSP(Z-N)
-	IF(ID(Z)-1)49594
	5 L=LNKR(CØNT(LNKL(CØNT(LNKR(Z)))+1)))
•	GØ TØ 3
-	4 L=LNKR(Z)
	1  SFOSR = CONT(1+1)
un	$Z = C \emptyset N T (L)$
6-116	N=ID(Z)-1
•	RETURN
	GØ TØ 3
LOR	END
X	
sto	
•	
<u>_</u>	
No Ilon	
NN-H	
XURIE	
	\$ETURN
	S=CRM1(C)
12	
10	
9	STOISSEEDT 0101
8	
7	
5	
4	
3	
2	

IT THE		
•		
3	ZZ SEQSL 0705	
3	ZZEØR	
0		
(m)	*FANDK0810	
cu .	*LDISKSEQSL 0705	- Permo 180
0	FUNCTION SEQSL(Z,N)	
	IF(ID(Z)-1)4,5,4	
N	5 L=LNKL(CØNT(CØNT(LNKL(CØNT(LNKR(Z)))+1)))	
0	GØ TØ 3	
	4 L=LNKL(Z)	
	3 IF(ID(CØNT(L))-1)1+2+1	
0	1 SEQSL=CONT(L+1)	
88	Z = CONT(L)	T DATE VIET
-		
	2 1 -1 NKL (CONT (CONT (1+1)))	CONVOID
	GØ TØ 3	TDOWOGRA
0	END	
×		T DOWOOTS
TO	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.	PDOW0018
0	C II.** I=000 POSITIVE. 2=EVEN. 3=000 NEGATIVE.	
	11=3-01+01\5*5	
7	EQUIAVIENCE (XAPIR)	
0		
1170		
	IA+R.J C.D.S.B.+S4 AIB DEFGUI AIBC 1	
0 8		
· · ·		
0		
	CALL LDUMP11+11	
	THANDERFIEL	
0		
0	MEANDKO810	
	222 CKX	
0	22.126	
-		
1		

	S'AMPLE PROGRAM + DATA	
	ZZJØB 29270126507387+215	
The		
	ZZFØRX	
	*FANDK0810	
	B=0.	
	L=LIST(B)	
	10 CALL LISTRD(L,5)	
	I=NUCELL(L)	
	CALL LDUMP(1,I)	
	CALL RCELL(I)	
-	END (122)	
1185		
· · · · ·	(123-456)	
	(A+0) (ADAS B +5 -4 AUBCDEEGHI ASBC)	
NOL T	FUNCTION LDUMP (N1,N2)	LDUM0010
ž	EQUIVALENCE (XW,IW)	LDUM0011
510	IT=2-N1+N1/2*2	LDUM0012
	C IT 1=ØDD PØSITIVE, 2=EVEN, 3=ØDD NEGATIVE.	LDUM0014
-	LL=LNKL(CØNT(-1))	LDUM0016
ME	LR=LNKR(CØNT(-1))	LDUM0018
	PUNCH 3, IT, N1, N2, LL, LR	LDUM0020
	3 FØRMAT(I1,5X,19HLDUMP ØF SLIP-CELLS,I6,7H THRU,I6,	LDUM0030
CIRIE	1 10H AVSL=,216)	LDUM0031
		LDUM0040
	60 10 (49090)911	LDUM0042
-	5 EXAMATICENO ADDRESS RE EIRST CELL MUST BE ROD AND DROLTIVE	
·	10 MITTED	
-	RETURN	
	4 DØ 2 I=N1.N2.2	
	LL = LNKL(CØNT(I))	1 DUM0080
12	LR=LNKR(CØNT(I))	LDUM0090
	1X = ID(CØNT(I))	LDUM0100
9	XW = CONT(I+1)	LDUM0120
	LDUMP=LDUMP+1	LDUM0122
· ·	2 PUNCH 1, I, I, LL, LR, IX, IW, XW	LDUM0130
6	1 FØRMAT (1X, 15, 3X, 5HLNKL=, 15, 3X, 5HLNKR=, 15, 3X, 3HID=, 12, 3X,	LDUM0140
5	1 6HDATUM=, I11,2X,2HØR,E16.10)	LDUM0150
🧐 a 🔜	RETURN	LDUM0160
3		

			- NR 10 2							
			1							
		EN	1D							LDUM0200
	)	. EI	- EMAGAS.	1. 4 d 1.	- Enkisell	68	104 9	DETONA 120000000	and the second	A SALE AND A SALE AND A SALE AND A
			1.44.1.4							
	1000		LANE		L. Market	1.3.2		A A A A A A A A A A A A A A A A A A A		1 2 3 3 4 2 2 2 2 2 4 2 4 1 4 1 4 1 4 1 4 1 4 1
			TAKE *							
1		28.	1,114,1 =	193	1.28 B	- 11	1074 0	DATOR: 1400000000	199	
		1.1	FUE	1	1013 3 0		105 9		1912	· · · · · · · · · · · · · · · · · · ·
	-	1 2 2	C. BIK L. Y	16.68	CHADE		THE M	00110N+ 0000000000	NEW	KENSASDAST LAR
	Maria I.		ENKL S							PHADUC PHAR . 16
	-	193	I PIAL T		Contraction of the	2.4	112		-	
									15W	
										*420000000000
		Tues a	- Have -	1.2.4	Contraction in		TOA O	Swither Kasaroar		HE SUCCOSOF - TR
		1								*D.00000002.20
										* 5300073087301
	-		TURNE S		THE REAL	144	The		ALC .	* 30 (() 13 2 3 4 C + 1 4
				-	THE DE					
	Mary Prod	1 1 2	TAXY C	191	Their to a					1120000000-10
		100	CINY T'A	100	THE R.	12.55	10= 0	DWINKE STISSESSION	Di H	1012302220410
					The Real					
		1 12 3		100	Carlos and					1 10 VI 10005 + 10
				1	Flat star	TRA	TOTAL OF	Dal Che Steraporte 10		1026404040404040
				2.7						
			TV2	TA	Clay 2 -					
		133	CHEFE	1.11	T IT I	TET	100	DETONS SECTORACION	No.	**************************************
		1.45	C LANGT OF	TRI	T THE R					7130001805+03
		193	CONCE	151	TINKAS	715	10+ 5	Detra-	NK	*550000000000
		1	THEFT	THE A	TIMER					
		TINT	CNETE	The	Chicke -	TIA	10= 1	DVIIMA - PDP220392	RIS	F#EE#29#3E+10
	-				THESE					*TOCOUTEDE+OP
			EMKI =							*C000000000
		LUY	THEFT	1993	TURSA		The S	Dalline -2	SE	********

OUTPUT FROM SAMPLE PROGRAM

rales -											
		LDUMP ØF	SLIP-	CELLS	1	THRU	669	Concession of	AVSL= 3	3 6	71
	1	LNKL=	667	LNKR=	7	ID=	2 D	ATUM=	2	ØR	•2000000E+01
	3	LNKL=	3	LNKR=	0	ID=	2 D	ATUM=	(	ØR	•0000000E-99
3	5	LNKL=	3	LNKR=	3	ID=	1 D	ATUM=	100001	ØR	.10000100E+06
	7	LNKL=	1	LNKR=	9	ID=	0 D.	ATUM=	4664554363	B ØR	•46645543E+10
	9	LNKL=	7	LNKR=	11	ID=	0 D	ATUM=	4956550000	ØR	•49565500E+10
	11	LNKL=	9	LNKR=	15	ID=	0 D.	ATUM=	5344645457	ØR	•53446454E+10
	13	LNKL=	23	LNKR=	19	ID=	2 D.	ATUM=	2	ØR	•2000000E+01
	15	LNKL=	11	LNKR=	25	ID=	1 D	ATUM=	1300013	B ØR	.13000130E+07
3	17	LNKL=	3	LNKR=	39	ID=	1 D.	ATUM=	1300013	8 ØR	•13000130E+07
	19	LNKL=	13	LNKR=	21	ID=	0 D.	ATUM=	5571000000	) ØR	•55710000E+10
	21	LNKL=	19	LNKR=	23	ID=	0 D.	ATUM=	230000000	ØR	.2300000E+10
	23	LNKL=	21	LNKR=	13	ID=	0 D.	ATUM=	5572000000	ØR	•55720000E+10
in and the second se	25	LNKL=	15	LNKR=	27	ID=	O D	ATUM=	5344645470	ØR	•53446454E+10
Ť.	27	LNKL=	25	LNKR=	29	ID=	0 D.	ATUM=	7071700000	ØR	•70717000E+10
	29	LNKL=	27	LNKR=	31	ID=	0 D.	ATUM=	4558644965	ØR	•45586449E+10
A STREET BOARD	31	LNKL=	29	LNKR=	33	ID=	0 D.	ATUM=	4153455543	ØR	•41534555E+10
2	33	LNKL=	31	LNKR=	37	ID=	0 D.	ATUM=	450000000	ØR	.4500000E+10
2	35	LNKL=	45	LNKR=	41	ID=	2 D.	ATUM=	2	ØR	•2000000E+01
CCK	37	LNKL=	33	LNKR=	47	ID=	1 D.	ATUM=	3500035	ØR	• 35000350E+07
sto	39	LNKL=	3	LNKR=	139	ID=	1 D.	ATUM=	3500035	ØR	•35000350E+07
	41	LNKL=	35	LNKR=	43	ID=	0 D.	ATUM=	676600000	ØR	•67660000E+10
-	43	LNKL=	41	LNKR=	45	ID=	0 D.	ATUM=	230000000	ØR	•2300000E+10
WE	45	LNKL=	43	LNKR=	35	ID=	O D.	ATUM=	4966000000	ØR	•49660000E+10
Lo I	47	LNKL=	37	LNKR=	49	ID=	0 D.	ATUM=	5344645470	ØR	•53446454E+10
-UN	49	LNKL=	47	LNKR=	51	ID=	0 D.	ATUM=	7071710000	ØR	•70717100E+10
<u> </u>	51	LNKL=	49	LNKR=	53	ID=	0 D,	ATUM=	4963000000	ØR	•49630000E+10
· · · · · · · · · · · · · · · · · · ·	53	LNKL=	51	LNKR=	55	ID=	O D	ATUM=	330000000	ØR	.3300000E+10
	55	LNKL=	53	LNKR=	57	ID= (	0 D.	ATUM=	2	ØR	•2000000E+01
	57	LNKL=	55	LNKR=	59	ID= (	0 D,	ATUM=	200000000	ØR	.2000000E+10
<b>9</b>	59	LNKL=	57	LNKR=	61	ID= (	O DI	ATUM=	5571000000	ØR	•55710000E+10
	61	LNKL=	59	LNKR=	63	ID= (	0 D,	ATUM=	100000000	ØR	.1000000E+10
	63	LNKL=	61	LNKR=	65	ID= (	O DI	ATUM=	5571000000	ØR	•55710000E+10
	65	LNKL=	63	LNKR=	67	ID= (	O Di	ATUM=	210000000	ØR	•2100000E+10
	67	LNKL=	65	LNKR=	69	ID= (	O DI	ATUM=	2	ØR	•2000000E+01
11	69	LNKL=	67	LNKR=	71	ID= (	0 D./	ATUM=	140000000	ØR	.1400000E+10
10	71	LNKL=	69	LNKR=	73	ID = (	D Di	ATUM=	2	ØR	•2000000E+01
9	73	LNKL=	71	LNKR=	75	ID= (	D DI	ATUM=	5344645470	ØR	•53446454E+10
8	75	LNKL=	73	LNKR=	77	ID = (	O Di	ATUM=	7071720000	ØR	•70717200E+10
7	77	LNKL=	75	LNKR=	79	ID = 0	O DI	ATUM=	430000000	ØR	•4300000E+10
6	79	LNKL=	77	LNKR=	81	ID= (	D DI	ATUM=	496300000	ØR	•49630000E+10
5	81	LNKL=	79	LNKR=	83	ID = (	D Di	ATUM=	30000000	ØR	.3000000E+09
A	83	LNKL=	81	LNKR=	85	ID= (	D DI	ATUM=	300000000	ØR	.3000000E+09

-

1									-	
0	85	LNKL=	83	LNKR=	87	ID= 0	DATUM=	1	ØR	.1000000E+01
	87	LNKL=	85	LNKR=	89	ID= 0	DATUM=	330000000	ØR	• 33000000E+10
	89	LNKL=	87	LNKR=	91	ID = 0	DATUM=	5644440000	ØR	.56444400E+10
9	91	LNKL=	89	LNKR=	93	ID= 0	DATUM=	5756624963	ØR	.57566249E+10
	93	LNKL=	91	LNKR=	95	ID= 0	DATUM=	4965450000	ØR	•49654500E+10
	95	LNKL=	93	LNKR=	97	ID = 0	DATUM=	230000000	ØR	.2300000E+10
0	97	LNKL=	95	LNKR=	99	ID= 0	DATUM=	2	ØR	•2000000E+01
	99	LNKL=	97	LNKR=	101	ID = 0	DATUM=	330000000	ØR	.3300000E+10
	101	LNKL=	99	LNKR=	103	ID = 0	DATUM=	4565455500	ØR	•45654555E+10
	103	LNKL=	101	LNKR=	105	ID= 0	DATUM=	230000000	ØR	.2300000E+10
	105	LNKL=	103	LNKR=	107	ID = 0	DATUM=	190000003	ØR	.3000000E+01
	107	LNKL=	105	LNKR=	109	ID = 0	DATUM=	3300000000	ØR	•3300000E+10
	109	LNKL=	107	LNKR=	111	ID = 0	DATUM=	5644440000	ØR	.56444400E+10
10	111	LNKL=	109	LNKR=	113	ID = 0	DATUM=	5545474163	ØR	•55454741E+10
	113	LNKL=	111	LNKR=	115	ID = 0	DATUM=	4965450000	ØR	•49654500E+10
	115	LNKL=	113	LNKR=	117	· ID= 0	DATUM=	30000000	ØR	.3000000E+09
	117	LNKL=	115	LNKR=	119	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
x	119	LNKL=	117	LNKR=	121	ID= 0	DATUM=	7071740000	ØR	•70717400E+10
	121	LNKL=	119	LNKR=	123	ID= 0	DATUM=	5353000000	ØR	.53530000E+10
ž	123	LNKL=	121	LNKR=	125	ID= 0	DATUM=	3300000000	ØR	.3300000E+10
s10	125	LNKL=	123	LNKR=	129	ID= 0	DATUM=	5355525300	ØR	•53555253E+10
0	127	LNKL=	137	LNKR=	133	ID= 2	DATUM=	586000002	ØR	.2000000E+01
	129	LNKL=	125	LNKR=	143	ID= 1	DATUM=	12700127	ØR	.12700127E+08
×	131	LNKL=	3	LNKR=	165	ID= 1	DATUM=	12700127	ØR	.12700127E+08
	133	LNKL=	127	LNKR=	137	ID= 0	DATUM=	4356556300	ØR	.43565563E+10
N	135	LNKL=	141	LNKR=	141	ID= 2	DATUM=	2	ØR	.2000000E+01
ALER.	137	LNKL=	133	LNKR=	127	ID= 1	DATUM=	13500135	ØR	.13500135E+08
	139	LNKL=	3	LNKR=	131	ID= 1	DATUM=	13500135	ØR	.13500135E+08
	141	LNKL=	135	LNKR=	135	ID= 0	DATUM=	-1	ØR	1000000E+01
	143	LNKL=	129	LNKR=	145	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
3	145	LNKL=	143	LNKR=	147	ID= 0	DATUM=	7071760000	ØR	.70717600E+10
	147	LNKL=	145	LNKR=	149	ID= 0	DATUM=	5359000000	ØR	•53590000E+10
	149	LNKL=	147	LNKR=	151	ID= 0	DATUM=	330000000	ØR	•3300000E+10
0	151	LNKL=	149	LNKR=	155	ID= 0	DATUM=	5355525900	ØR	.53555259E+10
	153	LNKL=	163	LNKR=	159	ID= 2	DATUM=	2	ØR	.2000000E+01
12	155	LNKL=	151	LNKR=	169	ID= 1	DATUM=	15300153	ØR	.15300153E+08
)"	157	LNKL=	3	LNKR=	211	ID= 1	DATUM=	15300153	ØR	.15300153E+08
	159	LNKL=	153	LNKR=	163	ID= 0	DATUM=	4356556300	ØR	.43565563E+10
9	161	LNKL=	167	LNKR=	167	ID = 2	DATUM=	2	ØR	.2000000E+01
0 *	163	LNKL=	159	LNKR=	153	ID= 1	DATUM=	16100161	ØR	.16100161E+08
7	165	LNKL=	3	LNKR=	157	ID= 1	DATUM=	16100161	ØR	.16100161E+08
6	167	LNKL=	161	LNKR=	161	ID= 0	DATUM=	-1	ØR	1000000E+01
0 5	169	LNKL =	155	LNKR=	171	ID= 0	DATUM=	5344645470	ØR	-53446454E+10
4	171	I NIKI -	160	I NKD-	170	10 0	DATUM	7071700000	on	

.

	State 111	THEF -	1 199	THEF						
	173	LNKL=	171	LNKR=	175	ID = 0	DATUM= 5764554	4348	ØR	.57645543E+10
	175	LNKL=	173	LNKR=	177	ID= 0	DATUM=	3	ØR	.3000000E+01
	177	LNKL=	175	LNKR=	179	ID= 0	DATUM= 2300000	0000	ØR	.2300000E+10
	179	LNKL=	177	LNKR=	181	ID= 0	DATUM= 4963000	0000	ØR	•49630000E+10
	181	LNKL=	179	LNKR=	183	ID = 0	DATUM= 2300000	0000	ØR	.2300000E+10
	183	LNKL=	181	LNKR=	185	ID= 0	DATUM= 5571000	0000	ØR	•55710000E+10
	185	LNKL=	183	LNKR=	187	ID= 0	DATUM= 2300000	0000	ØR	.2300000E+10
	187	LNKL=	185	LNKR=	189	ID = 0	DATUM= 5572000	0000	ØR	•55720000E+10
	189	LNKL=	187	LNKR=	191	ID= 0	DATUM= 2300000	0000	ØR	-2300000E+10
	191	LNKL=	189	LNKR=	193	ID= 0	DATUM= 5353000	0000	ØR	•53530000E+10
	193	LNKL=	191	LNKR=	195	ID= 0	DATUM= 2300000	0000	ØR	2300000E+10
	195	LNKL=	193	LNKR=	197	ID= 0	DATUM= 5359000	0000	ØR	•53590000E+10
	197	LNKL=	195	LNKR=	199	ID= 0	DATUM= 5344645	5470	ØR	•53446454E+10
10	199	LNKL=	197	LNKR=	201	ID = 0	DATUM= 7072700	0000	ØR	-70727000E+10
118	201	LNKL=	199	LNKR=	203	ID= 0	DATUM=	3	ØR	-3000000E+01
· · · · · · · · · · · · · · · · · · ·	203	LNKL=	201	LNKR=	205	ID= 0	DATUM= 4656595	5441	ØR	46565954E+10
	205	LNKL=	203	LNKR=	209	ID = 0	DATUM= 6300000	0000	ØR	-6300000E+10
	207	LNKL=	263	LNKR=	213	ID= 2	DATUM=	2	ØR	2000000E+01
No To	209	LNKL=	205	LNKR=	265	ID = 1	DATUM= 20700	0207	ØR	20700207E+08
×	211	LNKL=	3	LNKR=	287	ID = 1	DATUM= 20700	0207	ØR	-20700207E+08
01	213	LNKL =	207	LNKR=	215	ID= 0	DATUM= 4971000	0000	ØR	49710000E+10
· · · · · · · · · · · · · · · · · · ·	215	1 NKI =	213	LNKR=	217	ID= 0	DATUM= 2300000	0000	ØR	-23000000E+10
	217	INKI =	215	LNKR=	219	ID= 0	DATUM= 67000	0005	ØR	67000005E+08
	219	LNKL=	217	LNKR=	221	ID = 0	DATUM= 2300000	0000	ØR	2300000E+10
C	221	LNKL=	219	LNKR=	223	ID = 0	DATUM= 48534	4464	ØR	48534464E+08
INN	223	LNKL=	221	LNKR=	225	ID = 0	DATUM= 5457000	0000	ØR	54570000E+10
ER	225	LNKL=	223	LNKR=	227	ID= 0	DATUM= 5646000	0000	ØR	•56460000E+10
Con the second	227	LNKL=	225	LNKR=	229	ID = 0	DATUM= 6253495	5700	ØR	.62534957E+10
	229	LNKL=	227	LNKR=	231	ID= 0	DATUM= 2000000	0000	ØR	-2000000E+10
	231	LNKL=	229	LNKR=	233	ID= 0	DATUM= 4345535	5362	ØR	43455353E+10
	233	LNKL=	231	LNKR=	235	ID = 0	DATUM= 2300000	0000	ØR	-2300000E+10
	235	LNKL =	233	LNKR=	237	ID= 0	DATUM= 4976000	0000	ØR	49760000E+10
	237	LNKL =	235	LNKR=	239	ID= 0	DATUM= 2300000	0000	ØR	2300000E+10
	239	INKI =	237	INKR=	241	ID= 0	DATUM= 48000	0007	ØR	-48000007F+08
	241	INKI =	239	LNKR=	243	ID= 0	DATUM= 6348596	6400	ØR	-63485964E+10
12	243	LNKL =	241	LNKR=	245	ID= 0	DATUM= 2300000	0000	ØR	2300000E+10
····	245	LNKL =	243	LNKR=	247	ID= 0	DATUM= 4976000	0000	ØR	-49760000E+10
10	247	LNKL =	245	I NKR=	249	ID= 0	DATUM= 2300000	0000	ØP	-23000000E+10
9	249	LNKI =	247	LNKR=	251	ID= 0	DATUM= 5344645	5470	ØR	-53446454E+10
· · · · · · · · · · · · · · · · · · ·	251	LNKL =	249	LNKR=	253	ID= 0	DATUM= 7073700	0000	ØP	-70737000E+10
7	253	INKL =	251	LNKR=	255	ID= 0	DATUM=	1	ØID	-1000000E+01
6	255	INKI =	253	I NKR=	257	ID= 0	DATUM=	4810	ØP	-48100000E+04
5	257	LNKL =	255	I NKD-	250	ID= 0	DATUM= 4166628	5300	ØP	41656252E±10
4	621	LINKL	200	LUNN-	633	10- 0	UNION- 4100020	000	UR	+1000200LT10
3										

	250	INKI =	257	I NKP-	261	ID= 0	DATUM-	330000000	OID	3300000E+10
	261	INKI =	250	LNKR=	263	ID= 0	DATUM=	2300000000	ØP	- 23000000E+10
	263	I NKI =	261	I NKR=	207	ID= 0	DATUM=	490000000	ØR	-49000002F+09
	265	LNKI =	209	LNKR=	267	ID = 0	DATUM=	5344645470	ØR	-53446454E+10
	267	I NKI =	265	LNKR=	269	ID= 0	DATUM=	7073710000	ØP	-70737100E+10
	269	INKI =	267	I NKR=	271	ID= 0	DATUME	5344645457	ØP	-53446454E+10
	271	INKI =	269	I NKR=	273	ID= 0	DATUM=	3300000000	ØR	-3300000E+10
	273	LNKL =	271	I NKR=	275	ID= 0	DATUM=	00000000	ØR	-0000000E-99
	275	LNKL=	273	LNKR=	277	ID = 0	DATUM=	5344645470	ØR	•53446454E+10
	277	LNKI =	275	INKR=	279	ID= 0	DATUM=	7074700000	ØR	-70747000E+10
	279	INKI =	277	I NKR =	281	ID= 0	DATUM=	4756000000	ØR	47560000E+10
	281	INKI =	279	LNKR=	285	ID= 0	DATUME	6356000000	ØR	-63560000E+10
	283	INKI =	297	I NKR=	289	ID= 2	DATIM=	2220000000	ØR	2000000E+01
	285	LNKI =	281	LNKR=	299	ID = 1	DATIM=	28300283	ØR	28300283E+08
-	287	LNKI =	3	LNKR=	327	ID = 1	DATUM=	28300283	ØR	28300283E+08
	289	LNKL =	283	LNKR=	291	ID= 0	DATUM=	20300203	ØR	4000000E+01
	291	LNKL=	289	LNKR=	293	ID= 0	DATUM=	2300000000	ØR	+23000000E+10
	293	LNKL=	291	LNKR=	295	ID = 0	DATUM=	6	ØR	6000000E+01
	295	LNKL=	293	LNKR=	297	ID = 0	DATUM=	2300000000	ØR	=2300000E+10
	297	LNKL=	295	LNKR=	283	ID = 0	DATUM=	6	ØR	6000000E+01
-	299	LNKL=	285	LNKR=	301	ID= 0	DATUM=	2300000000	ØR	•2300000E+10
	301	LNKL=	299	LNKR=	303	ID= 0	DATUM=	4963000000	ØR	.49630000E+10
	303	LNKL=	301	LNKR=	305	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
	305	LNKL=	303	LNKR=	307	ID = 0	DATUM=	7074720000	ØR	.70747200E+10
	307	LNKL=	305	LNKR=	309	ID= 0	DATUM=	6	ØR	.6000000E+01
	309	LNKL=	307	LNKR=	311	ID = 0	DATUM=	5764554348	ØR	.57645543E+10
	311	LNKL=	309	LNKR=	313	ID = 0	DATUM=	0000005	ØR	5000000E+01
	313	LNKL=	311	LNKR=	315	ID = 0	DATUM=	5344645470	ØR	53446454E+10
	315	LNKL=	313	LNKR=	317	ID= 0	DATUM=	7074740000	ØR	.70747400E+10
	317	LNKL=	315	LNKR=	319	ID= 0	DATUM=	5	ØR	.5000000E+01
	319	LNKL=	317	LNKR=	321	ID= 0	DATUM=	4656595441	ØR	.46565954E+10
	321	LNKL=	319	LNKR=	325	ID= 0	DATUM=	630000000	ØR	.6300000E+10
1000	323	LNKL=	363	LNKR=	329	ID= 2	DATUM=	2	ØR	.2000000E+01
	325	LNKL=	321	LNKR=	365	ID= 1	DATUM=	32300323	ØR	.32300323E+08
	327	LNKL=	3	LNKR=	419	ID= 1	DATUM=	32300323	ØR	.32300323E+08
States and	329	LNKL=	323	LNKR=	331	ID= 0	DATUM=	480000650	ØR	.48000065E+09
	331	LNKL=	329	LNKR=	333	ID= 0	DATUM=	414445945	ØR	.41444459E+10
	333	LNKL=	331	LNKR=	335	ID= 0	DATUM=	6262000000	ØR	.62620000E+10
	335	LNKL=	333	LNKR=	337	ID= 0	DATUM=	5646000000	ØR	.56460000E+10
	337	LNKL=	335	LNKR=	339	ID= 0	DATUM=	4649596263	ØR	•46495962E+10
	339	LNKL=	337	LNKR=	341	ID= 0	DATUM=	4345535300	ØR	.43455353E+10
	341	LNKL=	339	LNKR=	343	ID= 0	DATUM=	5464626300	ØR	•54646263E+10
	343	LNKL=	341	LNKR=	345	ID= 0	DATUM=	4245000000	ØR	•42450000E+10
	345	LNKL=	343	LNKR=	347	ID= 0	DATUM=	5644440000	ØR	•56444400E+10

		02.00	CVKT =	3143	f Vik B=	213	10= 0	GW10MH	2644440000		
-	3	47	ENKL=	345	LNKR=	349	ID= 0	DATUM=	4155440000	ØR	.41554400E+10
	3	49	LNKL=	347	LNKR=	351	ID= 0	DATUM=	5756624963	ØR	.57566249E+10
	3	51	LNKL=	349	LNKR=	353	ID= 0	DATUM=	4965450000	ØR	•49654500E+10
-	3	53	LNKL=	351	LNKR=	355	ID = 0	DATUM=	200000000	ØR	.2000000E+10
	3	55	LNKL=	353	LNKR=	357	ID= 0	DATUM=	2053446454	ØR	20534464E+10
	3	57	LNKL=	355	LNKR=	359	ID= 0	DATUM=	-46	ØR	4600000E+02
-	3	59	LNKL=	357	LNKR=	361	ID= 0	DATUM=	1111111257	ØR	• 5700000E+02
	3	61	LNKL=	359	LNKR=	363	ID= 0	DATUM=	5654496363	ØR	.56544963E+10
	3	63	LNKL=	361	LNKR=	323	ID= 0	DATUM=	4544000000	ØR	.45440000E+10
-	3	65	LNKL=	325	LNKR=	367	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
	3	67	LNKL=	365	LNKR=	369	ID= 0	DATUM=	7074770000	ØR	.70747700E+10
	3	69	LNKL=	367	LNKR=	371	ID= 0	DATUM=	5945636459	ØR	•59456364E+10
	3	71	LNKL=	369	LNKR=	373	ID= 0	DATUM=	5500000000	ØR	.5500000E+10
in	3	73	LNKL=	371	LNKR=	375	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
.118	3	75	LNKL=	373	LNKR=	377	ID= 0	DATUM=	7075700000	ØR	.70757000E+10
- · ·	3	77	LNKL=	375	LNKR=	379	ID= 0	DATUM=	239999929 4	ØR	.4000000E+01
	3	79	LNKL=	377	LNKR=	381	ID= 0	DATUM=	4456000000	ØR	•44560000E+10
2	3	81	LNKL=	379	LNKR=	383	ID= 0	DATUM=	5164554342	ØR	.2000000E+01
COB CO	3	83	LNKL=	381	LNKR=	385	ID= 0	DATUM=	490000000	ØR	.4900000E+10
CK	3	85	LNKL=	383	LNKR=	387	ID= 0	DATUM=	330000000	ØR	.3300000E+10
STO.	3	87	LNKL=	385	LNKR=	389	ID= 0	DATUM=	5571000000	ØR	.55710000E+10
	3	89	LNKL=	387	LNKR=	391	ID= 0	DATUM=	230000000	ØR	.2300000E+10
-	3	91	LNKL=	389	LNKR=	393	ID = 0	DATUM=	5572000000	ØR	.55720000E+10
1	3	93	LNKL=	391	LNKR=	395	ID= 0	DATUM=	230000000	ØR	•2300000E+10
FOR	3	95	LNKL=	393	LNKR=	397	ID= 0	DATUM=	5309000002	ØR	.2000000E+01
NIN	* 3	97	LNKL=	395	LNKR=	399	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
RER	3	99	LNKL=	397	LNKR=	401	ID = 0	DATUM=	7076700000	ØR	.70767000E+10
ciui 🧑	4	01	LNKL=	399	LNKR=	403	ID = 0	DATUM=	5353000000	ØR	•53530000E+10
	4	03	LNKL=	401	LNKR=	405	ID= 0	DATUM=	3300000000	ØR	.33000000E+10
	4	05	LNKL=	403	LNKR=	409	ID= 0	DATUM=	5355525300	ØR	.53555253E+10
	4	07	LNKL=	417	LNKR=	413	ID = 2	DATUM=	2	ØR	.2000000E+01
	4	09	LNKL=	405	LNKR=	423	ID = 1	DATUM=	40700407	ØR	.40700407E+08
	4	11	LNKL=	3	LNKR=	445	ID= 1	DATUM=	40700407	ØR	.40700407E+08
	4	13	LNKL=	407	LNKR=	417	ID= 0	DATUM=	4356556300	ØR	•43565563E+10
13	4	15	LNKL=	421	LNKR=	421	ID= 2	DATUM=	2	ØR	.2000000E+01
	4	17	LNKL=	413	LNKR=	407	ID= 1	DATUM=	41500415	ØR	•41500415E+08
	4	19	LNKL=	3	LNKR=	411	ID = 1	DATUM=	41500415	ØR	•41500415E+08
10	4	21	LNKL=	415	LNKR=	415	ID= 0	DATUM=	490000000	ØR	.4900000E+10
	4	23	LNKL=	409	LNKR=	425	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
	4	25	LNKL=	423	LNKR=	427	ID= 0	DATUM=	7078700000	ØR	•70787000E+10
	4	27	LNKL=	425	LNKR=	429	ID= 0	DATUM=	5359000000	ØR	.53590000E+10
0	4	29	LNKL=	427	LNKR=	431	ID= 0	DATUM=	3300000000	ØR	.3300000E+10
	• 4	31	LNKL=	429	LNKR=	435	ID= 0	DATUM=	5355525900	ØR	•53555259E+10
4 .											

					4. H	Sector Sector			-	
	433	LNKL=	443	LNKR=	439	ID= 2	DATUM=	2	ØR	•2000000E+01
14.	435	LNKL=	431	LNKR=	449	ID= 1	DATUM=	43300433	ØR	•43300433E+08
	437	LNKL=	3	LNKR=	471	ID= 1	DATUM=	43300433	ØR	•43300433E+08
	439	LNKL=	433	LNKR=	443	ID = 0	DATUM=	4356556300	ØR	•43565563E+10
States.	441	LNKL=	447	LNKR=	447	ID= 2	DATUM=	2	ØR	•2000000E+01
	443	LNKL=	439	LNKR=	433	ID= 1	DATUM=	44100441	ØR	•44100441E+08
)	445	LNKL=	3	LNKR=	437	ID = 1	DATUM=	44100441	ØR	•44100441E+08
	447	LNKL=	441	LNKR=	441	ID= 0	DATUM=	4900000000	ØR	.4900000E+10
	449	LNKL=	435	LNKR=	451	ID= 0	DATUM=	5344645470	ØR	• 53446454E+10
	451	LNKL=	449	LNKR=	453	ID= 0	DATUM=	7079700000	ØR	•70797000E+10
	453	LNKL=	451	LNKR=	455	· ID= 0	DATUM=	496700000	ØR	•49670000E+10
	455	LNKL=	453	LNKR=	457	ID = 0	DATUM=	330000000	ØR	• 3300000E+10
	457	LNKL=	455	LNKR=	461	ID = 0	DATUM=	4944000000	ØR	•49440000E+10
in	459	LNKL=	469	LNKR=	465	ID = 2	DATUM=	2	ØR	• 2000000E+01
	461	LNKL=	457	LNKR=	475	ID = 1	DATUM=	45900459	ØR	•45900459E+08
	463	LNKL=	3	LNKR=	489	ID = 1	DATUM=	45900459	ØR	•45900459E+08
	465	LNKL=	459	LNKR=	469	ID = 0	DATUM=	4356556300	ØR	•43565563E+10
ž	. 467	LNKL=	473	LNKR=	473	ID = 2	DATUM=	2	ØR	•2000000E+01
For	469	LNKL=	465	LNKR=	459	ID = 1	DATUM=	46700467	ØR	•46700467E+08
ĊK	471	LNKL=	3	LNKR=	463	ID = 1	DATUM=	46700467	ØR	•46700467E+08
sto	473	LNKL=	467	LNKR=	467	ID = 0	DATUM=	490000000	ØR	•4900000E+10
	475	LNKL=	461	LNKR=	477	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
T	477	LNKL=	475	LNKR=	479	ID = 0	DATUM=	7170700000	ØR	•71707000E+10
W	479	LNKL=	477	LNKR=	481	ID = 0	DATUM=	676600000	ØR	.67660000E+10
Eor C	481	LNKL=	479	LNKR=	483	ID = 0	DATUM=	330000000	ØR	.3300000E+10
Nn-h	483	LNKL=	481	LNKR=	487	ID= 0	DATUM=	4356556300	ØR	•43565563E+10
RIE	485	LNKL=	495	LNKR=	491	ID = 2	DATUM=	2	ØR	•2000000E+01
COU	487	LNKL=	483	LNKR=	497	ID = 1	DATUM=	48500485	ØR	•48500485E+08
	489	LNKL=	3	LNKR=	559	ID = 1	DATUM=	48500485	ØR	•48500485E+08
	491	LNKL=	485	LNKR=	493	ID = 0	DATUM=	490000000	ØR	•4900000E+10
	493	LNKL=	491	LNKR=	495	ID = 0	DATUM=	100000000	ØR	•1000000E+10
	495	LNKL=	493	LNKR=	485	ID= 0	DATUM=	6766000001	ØR	.1000000E+01
4	497	LNKL=	487	LNKR=	499	ID = 0	DATUM=	5344645470	ØR	•53446454E+10
	499	LNKL=	4.97	LNKR=	501	ID = 0	DATUM=	7172700000	ØR	•71727000E+10
12	501	LNKL=	499	LNKR=	503	ID= 0	DATUM=	5344645457	ØR	•53446454E+10
11	503	LNKL=	501	LNKR=	505	ID = 0	DATUM=	330000000	ØR	•3300000E+10
0 10	505	LNKL=	503	LNKR=	507	ID= 0	DATUM=	5344645457	ØR	•53446454E+10
9	507	LNKL=	505	LNKR=	509	ID = 0	DATUM=	100000000	ØR	.1000000E+10
8	509	LNKL=	507	LNKR=	511	ID= Q	DATUM=	2100000601	ØR	.1000000E+01
	511	LNKL=	509	LNKR=	513	ID= O	DATUM=	5344645470	ØR	• 53446454E+10
	513	LNKL=	511	LNKR=	515	ID= 0	DATUM=	7172720000	ØR	•71727200E+10
	515	LNKL=	513	LNKR=	517	ID= 0	DATUM=	2	ØR	•2000000E+01
	517	LNKL=	515	LNKR=	519	ID = 0	DATUM=	5764554348	ØR	•57645543E+10

		254 6	Tyjkt +	1217							
-		521	INKI =	519	LNKR=	523	ID= 0	DATUM=	2300000000	ØR	-2300000E+10
		523	LNKL=	521	LNKR=	525	ID= 0	DATUM=	490000000	ØR	.4900000E+10
		525	LNKL=	523	LNKR=	527	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
		527	LNKL=	525	LNKR=	529	ID= 0	DATUM=	5353000000	ØR	•53530000E+10
		529	LNKL=	527	LNKR=	531	ID= 0	DATUM=	230000000	ØR	.2300000E+10
		531	LNKL=	529	LNKR=	533	ID= 0	DATUM=	5359000000	ØR	•53590000E+10
		533	LNKL=	531	LNKR=	535	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
		535	LNKL=	533	LNKR=	537	ID= 0	DATUM=	4967000000	ØR	.49670000E+10
	I CAN DESIGN	537	LNKL=	535	LNKR=	539	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
		539	LNKL=	537	LNKR=	541	ID= 0	DATUM=	4966000000	ØR	.49660000E+10
		541	LNKL=	539	LNKR=	543	ID= 0	DATUM=	2300000000	ØR	•2300000E+10
		543	LNKL=	541	LNKR=	545	ID= 0	DATUM=	6766000000	ØR	.67660000E+10
-		545	LNKL=	543	LNKR=	547	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
in		547	LNKL=	545	LNKR=	549	ID= 0	DATUM=	7173700000	ØR	.71737000E+10
118		549	LNKL=	547	LNKR=	551	ID= 0	DATUM=	1 48200441	ØR	.1000000E+01
A		551	LNKL=	549	LNKR=	553	ID= 0	DATUM=	4656595441	ØR	.46565954E+10
		553	LNKL=	551	LNKR=	557	ID= 0	DATUM=	630000000	ØR	.6300000E+10
E		555	LNKL=	649	LNKR=	561	ID = 2	DATUM=	2	ØR	.2000000E+01
OR OF		557	LNKL=	553	LNKR=	651	ID= 1	DATUM=	55500555	ØR	.55500555E+08
×	100 10000	559	LNKL=	3	LNKR=	5	ID= 1	DATUM=	55500555	ØR	.55500555E+08
stoc		561	LNKL=	555	LNKR=	563	ID= 0	DATUM=	67000001	ØR	.67000001E+08
		563	LNKL=	561	LNKR=	565	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
	The Render State	565	LNKL=	563	LNKR=	567	ID= 0	DATUM=	4975000000	ØR	.49750000E+10
×		567	LNKL=	565	LNKR=	569	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
Lon 👟		569	LNKL=	567	LNKR=	571	ID= 0	DATUM=	67000003	ØR	.6700003E+08
INN		571	LNKL=	569	LNKR=	573	ID= 0	DATUM=	230000000	ØR	.2300000E+10
RER		573	LNKL=	571	LNKR=	575	ID = 0	DATUM=	48535552	ØR	•48535552E+08
cont		575	LNKL=	573	LNKR=	577	ID= 0	DATUM=	530000000	ØR	.5300000E+10
	T REPORT OF	577	LNKL=	575	LNKR=	579	ID= 0	DATUM=	330000000	ØR	.33000000E+10
		579	LNKL=	577	LNKR=	581	ID = 0	DATUM=	2300000000	ØR	.2300000E+10
		581	LNKL=	579	LNKR=	583	ID = 0	DATUM=	4975000000	ØR	.49750000E+10
		583	.LNKL=	581	LNKR=	585	ID = 0	DATUM=	2300000000	ØR	•2300000E+10
		585	LNKL=	583	LNKR=	587	ID = 0	DATUM=	67000003	ØR	•6700003E+08
		587	LNKL=	585	LNKR=	589	ID = 0	DATUM=	2300000000	ØR	.2300000E+10
		589	LNKL=	587	LNKR=	591	ID= 0	DATUM=	48535552	ØR	• 48535552E+08
		591	LNKL=	589	LNKR=	593	ID = 0	DATUM=	5900000000	ØR	.5900000E+10
		593	LNKL=	591	LNKR=	595	ID= 0	DATUM=	330000000	ØR	.3300000E+10
10		595	LNKL=	593	LNKR=	597	ID = 0	DATUM=	2300000000	ØR	.2300000E+10
9		597	LNKL=	595	LNKR=	599	ID = 0	DATUM=	4975000000	ØR	•49750000E+10
		599	LNKL=	597	LNKR=	601	ID= 0	DATUM=	2300000000	ØR	•2300000E+10
7		601	LNKL=	599	LNKR=	603	ID= 0	DATUM=	6700003	ØR	.6700003E+08
6		603	LNKL=	601	LNKR=	605	ID= 0	DATUM=	2300000000	ØR	.2300000E+10
		605	LNKL=	603	LNKR=	607	ID= 0	DATUM=	48494403	ØR	•48494403E+08
4											

0										
	607	LNKL=	605	LNKR=	609	ID= 0	DATUM=	330000000	ØR	•3300000E+10
0	609	LNKL=	607	LNKR=	611	ID = 0	DATUM=	230000000	ØR	•2300000E+10
5	611	LNKL=	609	LNKR=	613	ID = 0	DATUM=	4972000000	ØR	.49720000E+10
2	613	LNKL=	611	LNKR=	615	ID = 0	DATUM=	230000000	ØR	•2300000E+10
V	615	LNKL=	613	LNKR=	617	ID = 0	DATUM=	6700003	ØR	•6700003E+08
	617	LNKL=	615	LNKR=	619	ID = 0	DATUM=	230000000	ØR	.2300000E+10
	619	LNKL=	617	LNKR=	621	ID= 0	DATUM=	5344645470	ØR	•53446454E+10
	621	LNKL=	619	LNKR=	623	ID = 0	DATUM=	7174700000	ØR	•71747000E+10
U	623	LNKL=	621	LNKR=	625	ID = 0	DATUM=	1	ØR	•1000000E+01
0	625	LNKL=	623	LNKR=	627	ID = 0	DATUM=	4844	ØR	•48440000E+04
	627	LNKL=	625	LNKR=	629	ID = 0	DATUM=	4163645400	ØR	•41636454E+10
	629	LNKL=	627	LNKR=	631	ID = 0	DATUM=	3300000000	ØR	.3300000E+10
0	631	LNKL=	629	LNKR=	633	ID= 0	DATUM=	230000000	ØR	•2300000E+10
1	633	LNKL=	631	LNKR=	635	ID = 0	DATUM=	4971710000	ØR	•49717100E+10
	635	LNKL=	633	LNKR=	637	ID = 0	DATUM=	230000000	ØR	.2300000E+10
0	637	LNKL=	635	LNKR=	639	ID = 0	DATUM=	6700002	ØR	.6700002E+08
	639	LNKL=	637	LNKR=	641	ID = 0	DATUM=	2300000000	ØR	.2300000E+10
W	641	LNKL=	639	LNKR=	643	ID = 0	DATUM=	48565902	ØR	•48565902E+08
O P	643	LNKL=	641	LNKR=	645	ID = 0	DATUM=	2300000000	ØR	.2300000E+10
CK	645	LNKL=	643	LNKR=	647	ID= 0	DATUM=	4571760000	ØR	•45717600E+10
STC	647	LNKL=	645	LNKR=	649	ID = 0	DATUM=	300000000	ØR	.3000000E+09
0	649	LNKL=	647	LNKR=	555	ID = 0	DATUM=	10	ØR	•1000000E+02
-	651	LNKL=	557	LNKR=	653	ID = 0	DATUM=	5344645470	ØR	•53446454E+10
WE	653	LNKL=	651	LNKR=	655	ID= 0	DATUM=	7175700000	ØR	•71757000E+10
C	655	LNKL=	653	LNKR=	657	ID = 0	DATUM=	5945636459	ØR	•59456364E+10
NUL-S	657	LNKL=	655	LNKR=	659	ID = 0	DATUM=	550000000	ØR	•5500000E+10
RIE	659	LNKL=	657	LNKR=	661	ID = 0	DATUM=	5344645470	ØR	.53446454E+10
000	661	LNKL=	659	LNKR=	663	ID= 0	DATUM=	7176700000	ØR	.71767000E+10
	663	LNKL=	661	LNKR=	665	ID = 0	DATUM=	4555440000	ØR	•45554400E+10
	665	LNKL=	663	LNKR=	667	ID = 0	DATUM=	5344645470	ØR	•53446454E+10
0	667	LNKL=	665	LNKR=	1	ID = O	DATUM=	7270700000	ØR	•72707000E+10
	669	LNKL=	0	LNKR=	0	ID = 0	DATUM=	0	ØR	•0000000E-99
13				1916 F	112.		1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.		1.1	
0 1										
							1000		1.1.1	
\$\$JOB 0000400002

## # # FORX

*FANDK0810 0324 LENGTH 59999 NEXT COMMON END OF COMPILATION EXECUTION

MAIN 1.	1200 00	1224 1	LUADED	
LIST	I1524	00574	LOADED	
NUCELL	Ī2098	01096	LOADED	
LDUMP	I3194	01750	LOADED	
RCELL	I4944	00612	LOADED	
INITAS	I5556	01028	LOADED	
LISTRD	I6584	03164	LOADED	
SETDIR	I9748	00374	LOADED	
IQUAL	20122	00618	LOADED	
SETIND	20740	00272	LOADED	
LNKR	21012	00110	LOADED	
CONT	21122	00158	LOADED	
IRALST	21280	00822	LOADED	
ID	22102	00158	LOADED	
STRIND	22260	00242	LOADED	
LNKL	22502	00122	LOADED	
KGET	22624	01140	LOADED	
NEWTOP	23764	00296	LOADED	
NEWBOT	24060	00296	LOADED	
TOP	24356	00362	LOADED	
KPUT	24718	00384	LOADED	
POPTOP	25102	00344	LOADED	
BOT_	25446	00362	LOADED	
LUCI	25808	00352	LUADED	
MTLLCT	20100	00002	LOADED	
ISTNAM	20402	00340	LOADED	
LUTDOT	27760	01010	LOADED	
NATET	50771	01014	LOADED	
DELETE	20729	01014	LOADED	
MAMTET	30682	00762	LOADED	
NAMISI	711.00	501-2C	LOADED	
DAGE	51450	300 51	LUADED	INITIAL
PAUL	y m y de	133 31	el ULLLO	1 FUL I I PA ba

ITIALIZED.

4-25-66

Kelleher and Sites

0.00

V15560	DEND: #+101
VISIOCENCI	HBE 4.987898.5-0
VICOS	BAC 41601R 17-0
YITAD	WATCEAS. DELATAGIUSSISSISSISSISSISSISSISSISSISSISSISSISS
VITRA	YES EVENEXT
HTTCH	PPROBELO -CL-TT
HITOOIP	HA EMPLEYSTO
HTTBOFAZ .	TH CHATTA-CHATT
0+99Acr	PE PLATT
WTT SAENT	BUL TEARENTY
UTTER.	TL FLITT + FERT
VITTREMI	PUP LETSSTA
VIIANCE	60 SFELL
PTOADLX5	68 65644,-CF-11
WINGARE IN	h2 \$5050+2+10
VIOVALVC	24.32 × 20
91000	DOKOGSTONCE-11
	D2C 1140.0
<b>WTARALENGT</b>	NAFCSS-M+2+FEMOLU+5+9+5+10+2+EMI-0+2+0+3
	BAGDO .LAKL .1-0
A1020	DC 0.987898.5-0
A10100	DS •*+101 Development of the second s
A1000****	ГИКГ 0630
*ID NOWBER	0630
*NYWE FUKE	
#STØRE REL	ØADA&LE
*ASSEMBLE	RELØCATABLE
*LIST CARD	
ZZ5P5	

LNKL 0630

-	
0	
0	
~	ZZ LNKL 0630
0	
20	ZZSPS
20	*LIST CARD
Le la	*ASSEMBLE RELØCATABLE
0	*STØRE RELØADABLE
	*NAME LNKL
	*ID NUMBER 0630
10	A1000**** LNKL 0630
	A1010Q DS ,*+101
	A1020 DC 6,987898,5-Q
0	A1030 DAC 6, LNKL , 7-Q
	A1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	A1050 DSC 17,0,0
0	A1060 DØRGQ-100
	A1070FAC DS ,2492
	A1080BETA DS ,2630
0	A1090FXZ DS ,3099
	A1100 DS 5
	A1110ENT AM *-1,5,10
0	A1120 TF CF+11,1-ENT
	A1130 BNF TF,CF+11
	A1140CF CF CF+11
O	A1150 TF CF+11,-CF-11
	A1160TF AM ENT-1,2,10
	A1170 TFL BETA,-CF-11
C R	A1180 TF FAC,FXZ
	A1190 MA FAC,BETA-5
	A1200 B7 1-ENT
0	A1210LENGTHDC 2,
	A1220 DEND
0	
-	
1	
	A REAL AND A
-	Kellesker and Siller
-	
-	

-	
•	
in the second	
	77 INKD 0422
	ZZ LNKK U03Z
-	775DC
	LLSPS
	*LIST CARD
-	*ACCEMPLE DELOCATABLE
-	ASSEMDLE RELOCATADEL
	*STORE RELOADADLE
-	*ID NUMBER 0632
-	B1000**** INKR 0632
	B10100 DS •*+101
	B1020 DC 6,987898,5-Q
-	B1030 DAC 6,LNKR ,7-Q
	B1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	B1050 DSC 17,0,0
	B1060 DØRGQ-100
	B1070FAC DS ,2492
	B1080FXZ DS ,3099
ž	B1090 DS 5
C.V.S	B1100ENT AM *-1,5,10
•	B1110 TF CF+11,1-ENT
	B1120 BNF TF,CF+11
1	BII30CF CF CF+II
•	$B1140 \qquad F  CF + II  F  F  F  F  F  F  F  F  F $
	BIISUIF AM ENI-1,2,10
IRIE	BIIGU IF FACSFX2
0	DIITO MA FACETCETI
-	DITULENOTION 29
	DIZUU UENU
-	
-	
•	

0	
Y Der	
0	
2	ZZ ID 0634
2	
20	ZZSPS
20	
-	*LIST CARD
0	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
- Inner	*NAME ID
0	*ID NUMBER 0634
民民	C1000**** 1D 0634
	C1010Q DS 9*+101
0	C1020 DC 6,987898,5-Q
1	Closo DAC 6, ID ,7-Q
	C1040 DVLC22-Q,5,LENGIH,2,8,2,10,5,ENI-6,5,0,30,0
0	C1050 DSC 17,0,0
	C1060 DØRGQ-100
Z	C1070FAC DS ,2492
O ê	CIO80BETA DS ,2630
ŏ	C1090FXZ DS ,3099
sto	C1100 DS 5
0	CITIOENT AM *-1,5,10
N.	CIISU BNF IF (CF+II
(House	
Nn-2	
RIET	CIIGUIF AM ENT-1,2,10
100	CIIPO IFL BETA,-CF-II
	CIIOU TE FACSEX
	C1190 BNF *+249BETA=1
0	CIIYZ AM FAC91910
	$C1194 \qquad DNF * + 249 DETA=2$ $C1196 \qquad AM = AC + 2 - 10$
-	C1200 P7 1-ENT
	C12101 ENGTHDC 2-1
	C1220 DEND
01	CIZZO DEND
0,10	
27	
-	
2	

•	
1	
•	
	ZZ CØNT 0636
A	
	ZZSPS
-	*LIST CARD
-	*ASSEMBLE RELOCATABLE
	*SIDRE RELDADADLE
	*NAME CONT
•	*ID NUMBER 0636
	D1000**** CØNT 0636
	D1010Q DS ,*+101
-	D1020 DC 6.987898.5-Q
•	D1030 DAC 6.CONT .7-0
10	$D1050$ DVL $C22=0.5 \pm 1$ ENGTH= 2 + 2 + 2 + 10 + 5 + ENT=6 + 5 + 0 + 30 + 0
•	D1050 DSC 17,0,0
	D1060 DØRGQ-100
	D1070FAC DS ,2492
	D1080FXZ DS ,3099
	D1082CALL BTM •*+11
0	
5	
•	
	DIIUDENI B CALL,,,TRANSFER VECTOR
2	DIIOSSLADR DS ,*-5
	D1109 AM ENT-1,5,10
No.	D1110 TF CF+11,1-ENT
ż	D1120 BNF IF •CF+11
- 5	DII30CE CE CE+11
	$P_{T}(P_{T}) = P_{T}(P_{T}) = P_{T}(P_{T})$
107- 5- 1ac. 18	DIIJOIP AM ENT-1,2,310
•	DI160 MA *+35,-CF-II
	D1165 MF *+23,-CF-11
751	DII70 BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
	D1175 TFL FAC,-99
-	D1180 B7 1-ENT
12	DII9000 DAC 6-INITAS.
ALL MARKED	
10	
9	
	DII96LENGTHDS ,QQ-1
•	D1200 DEND
· 4	

0	
0	
-	77 INITAS 0640
00	
00	ZZSPS
$\bigcirc$	
6	*LIST CARD
0	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
	*NAME INITAS
-	*ID NUMBER 0640
	E1000**** INITAS 0640
	F10100 DS •*+101
0	E1020 DC 6,987898,5-Q
0	F1030 DAC 6.INITAS.7-Q
	F1040 DVLC22-Q.5.LENGTH.2.8.2.10.5.ENT-6.5.0.30.0
00	F1050 DSC 17.0.0
0	E1060 $DØRGQ=100$
	E1060 D5. 2492
NRW C	ELOGABETA DS +2630
0.5	E106/EV7 DS +3099
OCK	E1064FX2 DS \$3097
15	
	E1080ENT AM #=1.7.10
- N	
- Pi	ELOOG TEM AZINEAL ELOOG FY THE TRANSFER VECTORS TO EXTERNALLY USED UTILITY ENTRY POINTS
40-2	E1100 TRNM-SLADR-
RIE	E1110CONT DC *=5
Cou	EIIIUCONI DS 9*=2
	EIIII TRNMSSLADRO
	EIIIZSE INDUS ,
	EIIIS IRNMISLADRO
	EIII4STRINDUS 9*=>
12.0	EIZZU**** CHAIN OF FREE SPACE - INITIALIZATION OF CHAINER
0	E1230 IFM LWRLM, 37980
	EI235 IFM UPRLM,40002
	EIZ40 IDM LWRLM99
0,	E1260 S UPRLM,LWRLM
	E1265 TF DISK+8,UPRLM-2
	E1267 TDM UPRLM-1
0	E1270 TDM UPRLM
	E1280 A UPRLM,LWRLM
	E1283 AM LWRLM,1,10
	E1286 TF DISK+13,LWRLM
	E1289 AM LWRLM, 19, 10
0	

•	<b>E1000</b>	TE AUCH AUCH
	E1292	IF AVSLL AVSLK
A	E1294	PEREARM CHAINING - MAY BE RE-USED BY ENTERING AT XXX
-	E1300****	AFTER SETTING BIKN AND RETN+6.
Varia:	E1301****	RTAMSLADR
	EISTUXAX	DIAMSLADA YTO
•	EISIZBLKN	DS 9*-4 TE EAC-EV1
3.5	E1315	
	E1910	TD FAC-49DENN
•	E1317 E1319	
	E1310	MA -AVSLA-AVSLR
-	C1313	TO MECITIO BLEN
•	E1320	UNTVMECT
12 -	E1330	WAITMEST
(1-0	EISBOLØØP	AM 99910910
•	E1390	
	E1400	1F -999FAC
No.	E1410	AM 99,10,10
02	E1430	IF -999FX2
¥UQ.	E1440	AM TAVELLEZIU
sto	E1450	DNE LOOD
•	E1460	SM 00.10.10
Γ.	E1402	TE -00.EV7
W	E1404	
0	E1470	CE RETA-2
NUL-2	E1472	DE A
RIEL	E1473SAVE	
03	E14/4	MM DEIA92910
	E1480	TE SAVE OR
	E1485	IF SAVE 990
•	E1490	WATTMEDZ
	E1500	SM -AVSLL92910
	E1501	MA FACISAVE
•	E1505RETN	B/ I-ENI
	E1525NEXT	
11	E1999	BI I-ENI EVTERNALLY CALLED UTILITIES
0 10	E2000****	EXTERNALLY CALLED UTILITIES
9	E2011*	CALLING SECHENCE - RIAM. SLADR. SLID-ADDRESS. 78 / RETHRNS
8	E2012****	MACHINE ADDRESS IN OD AFTER DASSIBLY THRNING A DISK DAGE.
0 7	E2013****	MACHINE ADDRESS IN 99 AFTER POSSIBLE TURNING A DISK FACE
	E2020	
	E2022SLADR	TE 999LWKLM
	E2025	DNCJNUTRP DNCJNU24
		BN(1*+24
		M 4

0		
-		
		BNC2*+48
D		PUT SWAP
20		A DISK+5, DISK+8
5		BNC1*+24
5		Н 5
	F2031NØTRP	BNE CHEK SLADR-1
The second second	E2032	
	E2032	
	EZUSSCHEN	DINV ADD
0	E2036	CF BACK+IU
	E2040ADD	A 98,SLADR-1
	E2045	BV ERR1
0	E2050	CM SLADR-5,,1011
	E2060PAGE	DS ,*
8	E2070	BE INCOR
6	E2080****	BLØCK NØT IN CØRE
0	F2290	Н 1
State Barris	EZADOLNCOR	C 99 ALIPRI M
N.	E2400INCON	
02	E2410	DH EKKI AM XIR JOID THDN DØGGIDLE CHECK BACK ØN
X	EZ411BACK	AM *+0,97910, TURN PESSIDLE CHECK DACK DN
C.	E2412	SF BACK+10
	E2413	BB2
	E2415ERR1	TNF NMS1+36,SLADR-1
	E2420ERRØR	RCTY
	E2430	WATYNMS1
	E2435MES	DS ,*-5
	E2440	CALLEXIT
	E3000MES1	DMES.A. PAGE 9 - (E)
8	E3001MES2	DMES.A.XXXXX SLIP CELLS INITIALIZED.(E)
	ESOLONMEL	DMES A DSELLOG-ADDRESS YXXX OLT OF RANGE(F)
	ESUIDINISI	INTERNALLY DEEEDENCED CONSTANTS
•	E3120****	INTERNALLY REFERENCED CONSTANTS
	E3I30LWRLM	DS 5, ADDRESS OF RIGHTMOST DIGIT OF PSEUDO-WORD NUMBER TROUDUR
	E3132UPRLM	US 5, ADDRESS OF LAST PSEUDO-WORD TEND OF SECTOR DEDCKT
	E3140DISK	DDA
	E3145SWAP	DD ,DISK
12	E3340AVSLL	DS 5, , PØINTERS TØ FIELD ADDRESSES ØF RIGHT AND LEFT
011	E3350AVSLR	DS ,LWRLM, LINKS ØF AVSL IN CØRE (SLIP CELL X0000).
10	F3440***	EXTERNALLY USED UTILITY ENTRY LINKAGES
a distanti	E3450SLADP	N/AD STADR
ang 8	E3450	DODG8-4
0,	E3400	
	E3470	DAG ( CANT
	E3680QQ	DAC DOCUME O
•	E3690	DVLC,5,CONT
	E2601	DAC CETIND

	E3692 DVLC,5,SETIND
	E3694 DAC 6,STRIND
	E3696 DVLC,5,STRIND
	E3700 DC 2,
	E3/IULENGTHDS ,QQ-I
	EST20 DEND
-	
9	
6	
4	
4	

0	
-	
0	ZZ SETIND 0646
ວ	
De -	ZZSPS
0	
9	*LIST CARD
-	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
	*NAME SETIND
-	*ID NUMBER 0646
	G1000**** SETIND 0646
	G1010Q DS ,*+101
-	G1020 DC 6,987898,5-Q
	G1030 DAC 6,SETIND,7-Q
	G1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
0	G1050 DSC 17,0,0
	G1060 DØRGQ-100
	G1070ID DSA
A S	G1080LNKL DSA
X	G1090LNKR DSA
TOC	G1100CELL DSA .
-	G1110 DC 1,'
	G1112CALL BTM ,*+11
	G1114INITASDS ,*-5
(Allo	G1120 DS 5
UNIF	G1122ENT B CALL,,,TRANSFER VECTØR
-43	G1123SLADR DS ,*-5
O BIO	G1130 TFM TF+6, ID-4
	G1140 AM TF+6,4,10
	G1150 AM ENT-1,5,10
0	G1160 TF CF+11,1-ENT
	G1170 BNF *+36,CF+11
	G1180CF CF CF+11
0	G1190 TF CF+11,-CF-11
	G1200TF TF ,CF+11
	G1210 AM TF+6,1,10
0	G1220 BNR ENT+24,-TF-6
	G1230 AM ENT-1,1,10
	G2010 MA *+35,-CELL
0	G2011 MF *+23,-CELL
	G2020 BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
	G2030 MA CELL.,99
•	G2040 BTM ,*+11,,FØRTRAN CALLING LINKAGE TØ EXTERNAL SUBPRØGKAM
4	G2050SETDIRDS ,*-5

•	
-	
and the second second	
-0-	
A DE CONTRA	G2060 DSA -ID
	G2070 DSA -LNKL
	G2080 DSA -LNKR
-	G2090CELL DSA
•	GZIYUQQ DAC 6,SEIDIR,
	G2200 DVLC,5,SETDIR
PER PER PAR	G2202 DAC 6, INITAS,
•	G2204 DVLC,5,INITAS
	G2210 DC 2,
	G2220LENGTHDS •QQ-1
	62230 DEND
•	OLLIV DEND
19	
•	
2	
×	
8	
in .	
•	
The second	
3	
ų,	
•	
12	
11-1-1	
0 10	
•	
6	
5	
4	
3	
2	
•	

-	
	ZZ STRIND 0648
0	
	ZZSPS
2	
so l	*LIST CARD
	*ASSEMDLE RELUCATADLE
	*STORE RELOADABLE
	*NAME STRIND
	*ID NUMBER 0648
	H1000**** STRIND 0648
	H1010Q DS ,*+101
-	H1020 DC 6,987898,5-Q
	H1030 DAC 6,STRIND,7-Q
	H1040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
-	H1050 DSC 17,0,0
	H1060 DØRGQ-100
	H1065FAC DS ,2492
DRM	H1070DATUM DSA
	H1080ICL DSA
	H1110 DC 1,
5	H1112CALL BTM +*+11
	HILLAINITASDS **=5
	H1120 DS 5
RM	H1125ENT B CALL TRANSFER VECTØR
	H1126SLADR DS **=5
-0-E	H1130 TEM TE+6.DATUM-4
Line and	H1140 AM TE+6.4.10
- Og	H1150 AM ENT-1.5.10
	$\frac{11100}{11100} IE (E+1) = 1 = ENT$
-	H1170 BNE *+36.CE+11
•	
	HI220 AM IFTO91910
	H1220 BNR ENT+24,-1F-0
0	H1230 AM ENT-1,1,1,10
	H2010 MA *+35,-1CL
	H2011 MF *+23,-ICL
0	H2020 BTAM-SLADR,, 78, CALL TO INTERNAL UTILITY ROUTINE
	H2030 TEL FAC,-DATUM
	H2040 TFL -99,FAC
•	H2180 B7 1-ENT
4	H2190QQ DAC 6, INITAS,

0 . H2192 H2194 DVLC,5,INITAS DC 2,' A H2196LENGTHDS ,QQ-1 6 H2200 DEND 0 0 0 -0 0 • 0 0 0 0 0 -0

-	
-	
-	
N	77 VCET 0472
	ZZ NGET 0072
57	
	*NAME KGET
0	*ID NUMBER 0672
0	L1030 DAC 6.KGET .7-Q
	HIGH DAC ORACLE I A
-	
-	
-	
-	
-	
-	
-	
-	
-	
-	
1	
0	
-	
0	
-	
-	
1.1.1.1.1	
-	

- ALLE	
	7700
	ZZSPS
1 1 1 2 7 1 S	ALLET CADD
	*LIST CARD *ACCEMPLE DELOCATADIE
	*ASSEMDLE RELUCATADLE
	*STØRE RELØADADLE
	*NAME KTEST
	11000**** KGET 0672
	11010Q DS ,*+101
	11020 DC 6,987898,5-Q
	DAC 6, KTEST , 7-Q
	11040 DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
	11050 DSC 17,0,0
	Ilo60 DØRGQ-100
	I1061FAC DS ,2492
	IIO62BUF DAS 89,, MAY RELØCATE TØ FØRTRAN BUFFER AT LØC 2521
	11063BETA DS \$2630
Σ	I1064STZERØDS ,2771
	11065FXZ DS ,3099
X	I1066**** =9, .=5, )=6, +=3, \$=7, *=8, -=2, /=11, ,=13, (=10, ==1, *=14.
10 million	I1067TABLE DVLC, 2,9, 4,, 2,5, 2,6, 10,, 2,3, 4,, 2,7, 2,8, 10,, 2,2, 2,11
	I1068 DVLC, 2,, 2,13, 2,10, 16,, 2,1, 2,14
	I1069PØINT DSA BUF+158
I The second	I1070INDV DS 2,, 06 ØR 10
ORN	11072 DGM
and the second s	IIO74NEXT DS ,PØINT-4, DTA ØR DCA
	I1075MES DMES, A, EXIT CALLED BY KGET(E)
and	II078INPT DSA
8	IIO80NCØDE DSA
	11110 DC 1.
and the second second	11120 DS 5
	LII30ENT TEM TE+6. INPT-4
-	11140 AM ENT-1-5-10
	11150 AM ENT 195910
12	11100 IF CFT1191-ENT
11	
10	
9	
00	
7	
8	11220 DNK ENT-1 1 10
5	11250 AM ENT-191910
4	
3	11310 TE BETA EV7
2	11910 IF DETASFAL

	11220	DNE DEC	TNDT
	11320	DNF REC	
0-	11340	P7 AUT	
	LI350PEC	BND NO	DED DRINT
0	11390KEG	TEM DOIL	
when?	11300	MA DEL	
	11590	A DEV	VIC9-INFT VIC-DEVIC
	11400	RD TVE	
-	11410	TEM IND	NV.10.10
	11420	P7 *+2	20
	LIAAOTYPE	TEM INC	20
	114401172	H BHE	F=1
	11//3×***	CLEAR F	RUFFER IMAGE IS 87 INPUT CHARACTERS (BUE BUE+172).
	11442888		BLANK (BUE+174), AND 1 RECORD MARK (BUE+176).
17.0	I1445PERE	TR BUE	F=1.ct7FR0=1
	1144JRERE	H BUE	= 19312LNO 1
	11450	TR BUE	F+86-ST7ERØ
NE	11420	H BUE	+174
	L1460ALL BL	BD *+2	24. INDV-1
DCK	11400ALLUL	PCTV	
ST	11402	GET NEX	ΥT ····································
-	11476	BD *+2	24 • INDV-1
	11477	BC4 RER	RE
all	11480	BI XM*+1	12.174(41)
E o	1148F	CM BUE	F-2(A1) • 10
. · ·	I148G	BNE NBI	X
INIC	11481	BCXM*-2	24 = 2(A1)
	11400	H 6	
	11481	B7 ALL	BI
-	I 148NNBLK	TOM BUE	F+2(A1)
-	L148PRKM	DC 1.	
	L148R	RNR *+2	24.BUF-2(A1)
	I148T	TEM BUE	=-2(A1),22,10
-	1148V	BCXM*=2	$24_{\bullet} = 2(\Delta 1)$
	LIASW	BNC3NOE	2FD+12
-11	1140W	H 7	
10	11494	RCTY	· · · · · · · · · · · · · · · · · · ·
9	11140	WATYBUE	
	1114F	B7 NØR	RED+12
7	L114GNØRED	AM POI	INT • 2 • 10
	1114.1	CM -PO	ZINT.22.10
	I 1 1 41	BNE NØR	RKM
4	11490	BNE ØKO	30 RMSW
	*****		

	I1492 WATYMES	
	IISOO CALLEXII	
	II5250KG0 SF RMSW	
	II530 B7 REG	
	IIS40NØRKM IF FAC,-PØINT	
	11550 CM FAC, 70, 10	
	11560 BL *+32	
	11565**** NUMERIC = 12	
	11570 IFM BETA,12,10	
	$11580 \qquad B7  DONE$	
	11590 CM FAC941910	
	LICOSXXXX ALDUARETIC - 04	
21-1-2	11000**** ALPHADE IIC - 04 11670 TEM BETA-04-10	
	LIGRO B7 DØNE	
	LIGSS**** NEITHER USE TARLE LØØK-UP	
1	110054444 NEITHER USE TABLE LOOK OF	2
	11700 BLX #12,FAC(A1)	
	11720 TE RETA TABLE(AL)	
	L2010DONE CE EAC-110	
	L2020DEVIC DS **	
	I2030RMSW DS **-2	
	12050KHSW 05 9 2 12040 CE BETA=1	
	12050 ME BETA, RMSW	
	12060 CE RMSW	
	12070ØUT TE -NCØDE.BETA	
	I2180 BSBBI-ENT	
	I2194LENGTHDC 1.	
	I2200 DEND	

4	ZZ LCNTR 0637
5	
20	ZZFØR
0	*EANDK 3810
-	*LDISKLCNTR 0637
	FUNCTION LENTR (LST)
	LCNTR=LNKR (CØNT (LST+1) )
-	RETURN LCN10030
	END LCN10040
-	
-	
•	
- NHO	
•	
-We	
- INIC	
IER-	
On	
tealtre	
-	
•	
13	A MANDARA BRITISH A TATA
-	
10	
9	
•	
-	
3	
•	

•		
•		
		and the second states of the second
-		
	77 LCTNAM 0639	
	LZ LSTNAM 0000	
A	77EOR	
-		
	*FANDK0810	
	*IDISKISTNAM0639	
	FUNCTIØN LSTNAM (LST)	LSTN0010
	LSTNAM=LNKL (CØNT (LST+1) )	LSTN0020
	CALL SETDIR (0,LSTNAM,LSTNAM,LSTNAM)	LSTN0030
	RETURN	LSTN0040
	END	LSTN0050
•		
H.		A STATE OF A
0		
ž		
STO		
•		
F		
0-40		
•		
•		
•		
	8	
	s and the second s	
•		
	a	
•		
Sec. 2		

-	
0	77 INTLBL 0649
S	
200	ZZFØR
0	
6	*FANDK0810
	*LDISKINTLBL0649
	FUNCTION INTUBL (I)
	END INTLOOSO
	END
-	
-	
0	
•	
-	
-	
-	
-	
0	
•	

•		
A STREET		
•		
1		
	ZZ FLILBL UODI	
A	77600	
•	ZZFOR	
	*FANDKO810	
	*LDISKFLTLBL0651	
•	FUNCTION FLTIBL (X)	FLTL0010
Less.	FITIBLEX	FLTL0020
	RETURN	FLTL0030
-	END	FLTL0040
1185		
2		
ž		
sto		
• •	and the second	
1	A State of the Physics of the State of the	
	And the second	
-		
-		
12		
10		
9		
5		
0		
2		
•		

-		
0	77 NUCELL 0653	
5	ZZ NUCLL 0000	
200	ZZFØR	
0		
S	*FANDK0810	
	*LDISKNUCELL0653	NUCCARIA
	FUNCTION NUCELL (X)	NUCEODIO
-	K=1	NUCE0020
	$\frac{10}{10} M = LNKR(CONT(-1))$	NUCEODSO
	$1F(M) \ge 9492$	NUCE0040
	4 00 10 (0,07, K	NUCEOOGO
	CALL INITAS (X)	NUCEO070
	GØ TØ 10	NUCEOD80
	8 TYPE 1	NUCE0090
-	1 FØRMAT(18HNØ MØRE FREE SPACE)	NUCEO100
	CALL EXIT	NUCEO110
-	2 IF(ID(CØNT(M))-1)12,14,12	NUCE0120
	14 CALL IRALST (CØNT(M+1))	NUCEO130
	2 CALL SETIND (-1,-1,LNKR(CONT(M)),-1)	NUCEO140
	CALL STRIND (0,M)	NUCE0150
Real Property	CALL STRIND (0,M+1)	NUCE0160
	NUCELL=M	NUCEO170
•	RETURN	NUCEO180
No.	END	NUCEDIAO
-		
•	and the second of the second o	
•		
	8	
-		
	3	
	2	
-		

•	
•	
-	
	77 RCELL 0655
	ZZFØR
	*FANDKU81U
•	SUBROUTINE RCELL (CELL) RCELOOIO
	CALL SETIND (-1,-1,CELL,LNKL(CØNT(-1))) RCEL0020
	CALL SETIND (-1,CELL,-1,-1) RCEL0030
	CALL SETIND (-1,-1,0,CELL) RCEL0040
	RETURN RCEL0050
•	END
-	
ž	
OCK	
ST	
-	
1	
NU-1	
CRIE	
•	
•	
•	
0	
•	

•			
	ZZ	MTLIST 0657	
	77F01	3	
	*FAN	DK0810	
	*LDI:	SKMTLIST0657	
A Street		FUNCTION MTLIST (LST)	MTLIOOIO
		L=LØCT(LST)	MTLI0020
•		IF(LISTMT(L)) 2,4,2	MTLI0030
	2	LR=LNKR(CØNT(L))	MTL10040
		LL = LNKL(CONI(L))	MTL 10050
•		CALL SETIND $(-1) = 1 + R = 1 \times K + (CQNT(-1))$	MTL 10070
10 00		CALL SETIND(-1.LL11)	MILIOOBO
		CALL SETIND(-1,-1,0,LL)	MTL10090
-	4	MTLIST=L	MTLIOIOO
		RETURN	MTLIOIIO
		END	MTLI0120
×			
100			
•			
• i			
8			
0			
11		The second s	
• 10			
	and the second		
7			
6			
•			
4			
3			
•			
•			

-		
-		
-		
1		
-9-	ZZ IRALST 0659	
	ZZFØR	
	*EANDY OR 10	
-	*LDISKIRALST0659	
-	FUNCTIØN IRALST (LST)	IRAL0010
	L=LØCT(LST)	IRAL0020
	CALL SETIND(-1,-1,-1,) CNTR(L) - 1, L+1) $IPALST = (CNTR(L))$	IRAL0040
	IF(IRALST)2,4,92	IRAL0050
	4 CALL MTLIST(L)	IRAL0060
	N=LSTNAM(L)	IRAL0070
811-8		IRAL0080
•	CALL SETIND(1,-1,-1,NEW)	IRAL0100
	CALL STRIND (N.NEW)	IRALOIIO
	CALL RCELL(NEW)	IRAL0120
2CK	8 CALL KCELL(L)	IRAL0150
a la	END	IRAL0150
-		
Į		
•	the same an action arrange solar arrange of the same solar arrange are solar and	
40-43		
C C C C C C C C C C C C C C C C C C C		
•	A Start The Destate of the Start Sta	
	I A REAL PROPERTY AND A RE	
	12	
-	IL THE REAL PROPERTY OF THE REAL PROPERTY AND	
-		
	8	
	6	
-	2	

•			
-			
•	77		
	22	LUOMP 0001	
	ZZEØR		
	227 211		
	*FAND	K0810	
-	*LDIS	KLDUMP 0661	
		FUNCTION LDUMP (N1,N2)	LDUMOOIO
		IT=2-N1+N1/2*2	LDUM0012
	C	IT 1=0DD POSITIVE, 2=EVEN, 3=0DD NEGATIVE.	LDUMO014
-		LL=LNKL(CØNT(-1))	LDUMO016
14 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	the second second second	LR=LNKR(CØNT(-1))	LDUMO018
-		PUNCH 3.IT.NI.N2.LL.	LDUMODZO
-	3	FØRMAT(11.5X.19HLDUMP ØF SLIP-CELLS.16.7H THRU.16.	LUUM0030
8		1 10H AVSL=,216)	LDUM0031
-		LDUMP=0	LDUM0040
•		GØ TØ (4.6.6) . IT	LUUM0042
	6	PUNCH 5	
A A A A A A A A A A A A A A A A A A A	5	FORMATIGSHO ADDRESS OF FIRST CELL MUST BE ODD AND POSITIVE-	
• 2	-	1P (MITTED)	
CCK		RETURN	
810	1.		LOUMOOSO
•	4		
			LDOMOUSO
1			LDUM0090
•		IX = ID(CONT(I))	LDUMO100
		IW = INILBL(CONI(I+1))	LDUMOIIO
A HERE		XW=CØN1(1+1)	LDUMOIZO
C un		LDUMP=LDUMP+1	LDUM0122
	2	PUNCH 1, I, LL, LR, IX, IW, XW	LDUM0130
	1	FØRMAT (1X,15,3X,5HLNKL=,15,3X,5HLNKR=,15,3X,3HIU=,12,3X,	LDUM0140
•		1 6HDATUM=, I11, 2X, 2HØR, E16.10)	LDUM0150
		RETURN	LDUM0160
		END	LDUM0200
0			
12			
10			
· ·			
7			
6			
5			
- A			

-		
-		
-		
	ZZ IQUAL 0665	
A		
•	ZZFØR	
	*FANDK0810	
•	*LDISKIQUAL 0665	10040010
	FUNCTION IQUAL (K,L)	19040010
	$4  \text{IF}(\text{LNKR}(K) - \text{LNKR}(L)) 2 \cdot 6 \cdot 2$	IQUA0030
-	6 IF(ID(K)-ID(L))2,8,2	IQUA0040
	C8 IQUAL=K-L	IQUA0050
	8 IQUAL=0	IQUA0050
		IQUA0070
	RETURN	IQUA0080
-	END	IQUA0090
MX		
1001		
•		
-		
A NUMBER		
- Interest		
TER-		
0		
•		
•		
	3	

-

•		
00	ZZ LISTMI 0667	
55	7.7 E (AD	
<b>**</b> •	ZZFØR	
10	*FANDKO810	
	*I DISKLISTMT0667	
•	FUNCTIØN LISTMT (LST)	LISTOOIO
and the second second second	L=LØCT(LST)	LIST0020
	LISTMT=IQUAL(CØNT(L),CØNT(LNKR(CØNT(L))))	LIST0030
	RETURN	LIST0040
The Part of the Pa	END	LIST0050
£		
100		
•		
1		
e al		
- IN		
•		
•		
	A REAL FILTER LETTER DESERT	
•		
	A CARACTER AND	ALL AND ADD ALL ALL ALL ALL ALL ALL ALL ALL AL
11		
10		
7		
6		
4		
a		
2		
THE REAL PROPERTY OF		

•	
•	
-0	
	ZZ NAMTST 0669
A	ZZFØR
	*FANDK0810
•	FUNCTION NAMIST (L)
	IF(LNKL(L)-LNKR(L))2,4,2 NAMT0020
	4 IF(ID(CØNT(L))-2)2,6,2 NAMT0030
	6 NAMTST=IQUAL(CØNT(LNKR(CØNT(LNKL(CØNT(L))))),CØNT(L)) NAMT0040
-	2 NAMTST=1 NAMT0070
	RETURN NAMTOO80
	END NAM10090
•	
U U	
X DO	
ts .	
W	
ER-UI	
O Dana	
-	
•	
9	
6	
7	
•	
3	
<b>a</b> ²	

O	ZZ LØCT 0671
0	
60	ZZFØR
0	
9	*FANDK0810
	*LDISKLØCT 0671
	FUNCTION LOCT (LST) LOCTODIO
	IF(NAMTST(LST))2,4,2 LØCT0020
-	4 LØCT=LST LØCT0030
	RETURN LØCT0040
	2 TYPE 1 LOCTO050
	1 FØRMAT(16HØPERAND NØT LIST) LØCT0060
	CALL EXIT
	END LØCT0080
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE D.P.K.
-	
1. A. A.	
0	
1	
212	
1000	
-	

•	
	ZZ LIST 0675
A	77500
•	ZZFØR
	*EANDY 2810
	*LDISKLIST 0675
-	FUNCTION LIST(K)
	LIST=NUCELL(Z)
-	CALL SETDIR(0,LIST,LIST,LIST)
	CALL SETIND(2,LIST,LIST,LIST)
	IF(IQUAL(K,9))2,1,2
	2 CALL SETIND(-1,-1,1,LIST+1)
	K=LIST
	1 RETURN
	END
1	
-	
-	
1	

	77 TAP 0677
6	22 IOF 0011
100	77 FØR
<b>—</b>	LLIDIN
20	*EANOKOGIO
	*FANDKUOIU
•	FUNCTION TODADA
	$\frac{10P=C0NI(LNKR(C0NI(L0CI(P)))+1)}{10P=C0NI(LNKR(C0NI(L0CI(P)))+1)}$
	RETURN
	END
1.2.	
•	
2	
Č.	
5	
•	
•	
NO.	
Ha la	
The second second	
•	
12	
· · · · · · · · · · · · · · · · · · ·	
10	
9	
7	
5	
3	
2	
•	
A STATE OF THE STATE OF	
Distance in the	
•	

•	
-0-	77 Par 0670
	ZZ BØT 0679
ê	ZZFØR
	* EANDY 0.910
-	*LDISKBØT 0679
	FUNCTION BOT(P)
	BØT=CØNT(LNKL(CØNT(LØCT(P)))+1) RETURN
-	END
-	
-	
-	
-	

16		
	ZZ	PØPTØP 0681
1944	775	
	ZZF	ØR
	*FA	NDK0810
	*LD	ISKPØPTØP0681
		FUNCTION POPTOP(P)
		RETURN
		END
	A STATE	

•	
TO	
•	
-	
F.	ZZ PØPBØT 0683
A	77500
•	ZZFØR
	*FANDK0810
-	*LDISKPØPBØT0683
	FUNCTION POPBOT(P)
	POPBOT=DELETETEINKETCONTILOCTIP/T/T
	END
No.	
0	
ž	
e s	
ock	
- Is	
-	
Ţ	
e se	
R-UN	
DURIE	
-	
12	
1	
6	
-	
4	
3	ZZ NEWTØP 0685
---------	--------------------------
0	77 500
0	ZZFØR
0	*EANDKORIO
-	*I DÍSKNEWTØP0685
•	EURICIAN NEWTOP(P.Q)
	NEWTØP=NXTRGT(P+LØCT(Q))
	RETURN
	END
-	
0	
-	
-	
-	
and and	
i i	
•	
0	
•	
•	
•	
-	
•	

-	
•	
4	ZZ NEWBØT 0687
A	22502
0	ZZFØR
	*FANDK0810
-	*LDISKNEWBØT0687
	FUNCTION NEWBOT(P,Q)
	NEWBØT=NXTLFT(P,LØCT(Q))
	RETURN
	END
-	
9	
•	
A NA	
100	
-	
<b>NIFO</b>	
-	CHERTER STREET
i al	
U IC	
-	

-	
d.	ZZ NXTRGT 0689
0	
0	ZZFØR
0	
0	*FANDK0810
	*LDISKNXTRGT0689
	FUNCTION NXTRGT(M.A)
	IR=NUCFIL(Z)
-	NYTPCT-IP
	CALL SETIND(-1 ID -1 (D)
	CALL SETIND(-1, I, I, ID, A)
	CALL SETIND(-1)-1)IK(A)
1	CALL SETIND(0,A,LK,IK)
	IF(NAMISI(M))1,2,1
· •	2 CALL SETIND(1,-1,-1,IR)
	CALL SETIND(-1,-1,LCNTR(M)+1,M+1)
	1 CØNTINUE
	CALL STRIND(M, IR+1)
	RETURN
00	END
in the	
•	
4	
HL H	
-	
-	
-	
10	
•	
•	
4	
-	
-	

		-
•		
-0-		
	ZZ NXTLFT 0691	
A	ZZFØR	
	*FANDK0810	
0	*LDISKNXTLFT0691	
	FUNCTION NXTLFT(M,A) $II = NII(EII (7)$	
-	NXTLFT=IL	
	LL=LNKL(CØNT(A))	
	CALL SETIND(-1,-1,IL,LL)	-
	CALL SETIND(-1,IL,-1,A)	
	IF(NAMTST(M))]=2=1	
10	2 CALL SETIND(1,-1,-1,IL)	
	CALL SETIND(-1,-1,LCNTR(M)+1,M+1)	
N	1 CALL STRIND(M,IL+1)	-
<ul> <li>● 2</li> </ul>	END	
QCK	END	
5		
- N		-
all of the		
10-23		
- Internet		
•		
		-
0		
12		
10		
9		
<b>e</b>		
5		
<b>•</b> .		
•		

	ZZ	LISTRD 0695
	ZZFØR	
-	*FAND	K0810
	*LDIS	KLISTRD0695
-		FUNCTION LISTRD(LST, IN)
and the second		EQUIVALENCE (NUM WORD)
		CALL KGET(-1)
/		I CLIB=1
	10	ICHAD=KCET/IN. ICODE)
	10	
	20	
	20	CALL NEWTODULCT LICTICTACKAA
	4.0	LOEL SHED
	40	IDELSW=0
		NUM=0
		IFLTSW=0
No.		ISINSW=0
×	50	ICHAR=KGET(IN,ICØDE)
100		IF (ICØDE) 51,51,54
10	C END	ØF LIST
	51	IF(IDELSW)53,53,52
F	52	CALL NEWBØT(WØRD, TØP(STACK))
	53	CALL IRALST(STACK)
R		RETURN
17	54	GØ TØ (110,120,130,140,150,160,170,180,190,200,210,220,230), ICOUL
	C FOU	AL SIGN
No.	110	IE ( IDE   SW ) 112, 112, 111
	110	CALL NEWBOT (WORD, TOPISTACK))
	112	NUM-0
	112	CALL KOUT (ICHAD, WORD, TODISTACK))
		CALL NEUPATIWARD TARLETACKIN
		CALL NEWDØI(WØRD)IØPISIACN//
,	c	
	C MINI	US SIGN
12	120	IF(IDELSW)121,121,111
	121	IDELSW=1
10		ISINSW=-1
9		CALL KPUT(ICHAR, WØRD, TØP(STACK))
8		GØ TØ 50
7	C PLUS	SSIGN
6	130	IF (IDELSW)131,131,111
5	131	IDEL SW=1
·		ISINSW=1
3	The second second second	

•	
1	
-9-	CALL KPUTLICHAR, WORD, TOPISTACK))
	GØ TØ 50
A	CLETTER
-	140 IDELSW=1
	CALL KPUT(ICHAR, WORD, TOP(STACK))
0	GØ TØ 50
	C DECIMAL PØINT
	150 IF(IDELSW)112,112,152
0	152 IF(IDELSW)153,111,111
	153 WØRD=NUM
	SIGN=ISIGN
0	IFLTSW=1
	DEC=10.
	GØ TØ 50
	C RIGHT PAREN
	160 IF(IDELSW)162,162,161
	161 CALL NEWBØT(WØRD, TØP(STACK))
- No.	162 CALL PØPTØP(STACK)
Y Y	I SUB=I SUB-1
	IF(ISUB)163,163,40
	163 CALL IRALST(STACK)
	RETURN
	C DØLLAR SIGN
e e	
N	CASTERIK
S S S S S S S S S S S S S S S S S S S	
	C BLANK
	190 IPTIDELSWIDU, JODISTACKI)
	CALL NEWDOT (WORD, TOP (STACK))
0	CLEET DAREN
	C LEFT PAREN 200 16/1061 CW1202 202 201
	200 IFTIDELSW/20292029201 201 CALL NEWPOTIMORD TODICTACKIN
0	201 CALL NEWDOT(WORD) TOP(STACK))
	CALL NEWBAT(LIST(0), TAP(STACK))
	CALL NEWTOP(BOT(TOP(STACK)) STACK)
0 10	GØ TØ 40
	C SLASH
0	210 GØ TØ 110
7	C NUMBER
6	220 IF(ISINSW)223,221,223
0.5	221 IF(IDELSW)222,222,224
4	222 ISINSW=1
	223 ISIGN=ISINSW
02	

-	
-	
	LCINCH-0
-	
0	IDELSW=1
00	IFLTSW=-1
0	NUM=(ICHAR-70)*ISIGN
S	GØ TØ 50
	224 IF(IFLTSW)225.140.226
	225 NUM=10*NUM+(ICHAR=70)*ISIGN
-	226 VAL=ICHAR-70
	WØRD=WØRD+SIGN*VAL/DEC
	DEC=DEC*10.
-	GØ TØ 50
0	CCAMMA
	220 60 10 110
0	END
14 2 6	
A R	
0.5	
C N	
1000	
an à	
ale.	
-	
-	
•	
10	
9	
4	
- 2	
•	
0	
Tear	

<ul> <li>ZZ DELETE 0693</li> <li>ZZFØR</li> <li>*FANDK0810</li> <li>*FUDISKDELETE0693</li> <li>*LDISKDELETE(K)</li> <li>FUNCTIØN DELETE(K)</li> <li>IF(ID(CØNT(K))-2)1,2,1)</li> <li>2 PRINT 100</li> <li>100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)</li> <li>DELETE=0.</li> <li>RETURN</li> <li>1 DELETE=CØNT(K+1)</li> </ul>	
ZZ DELETE 0693 ZZFØR *FANDKO810 *LDISKDELETE0693 FUNCTIØN DELETE(K) IF(1D(CØNT(K))=2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)	
<pre>ZZ DELETE 0693 ZZFØR  *FANDK0810 *LDISKDELETE0693 FUNCTIØN DELETE(K) IF(ID(CØNT(K))-2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)</pre>	
<ul> <li>ZZ DELETE 0693</li> <li>ZZFØR</li> <li>*FANDK0810</li> <li>*LDISKDELETE0693</li> <li>FUNCTIØN DELETE(K)</li> <li>iF(ID(CØNT(K))=2)1,2,1)</li> <li>2 PRINT 100</li> <li>100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)</li> <li>DELETE=0. RETURN</li> <li>1 DELETE=CØNT(K+1)</li> </ul>	
ZZ DELETE 0693 ZZFØR *FANDKO810 *LDISKDELETE0693 FUNCTIØN DELETE(K) IF(ID(CØNT(K))=2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)	
<ul> <li>ZZFØR</li> <li>*FANDK0810</li> <li>*LDISKDELETE0693</li> <li>FUNCTIØN DELETE(K)</li> <li>IF(ID(CØNT(K))-2)1,2,1)</li> <li>2 PRINT 100</li> <li>100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)</li> <li>DELETE=0.</li> <li>RETURN</li> <li>1 DELETE=CØNT(K+1)</li> </ul>	
<ul> <li>ZZFØR</li> <li>*FANDK0810</li> <li>*LDISKDELETE0693</li> <li>FUNCTIØN DELETE(K)</li> <li>IF(ID(CØNT(K))-2)1,2,1</li> <li>2 PRINT 100</li> <li>100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)</li> <li>DELETE=0.</li> <li>RETURN</li> <li>1 DELETE=CØNT(K+1)</li> </ul>	
<pre>*FANDK0810 *LDISKDELETE0693 FUNCTIØN DELETE(K) IF(ID(CØNT(K))-2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)</pre>	
<pre>*FANDK0810 *LDISKDELETE0693 FUNCTIØN DELETE(K) IF(ID(CØNT(K))-2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)</pre>	
<pre>*LDISKDELETE0893 FUNCTIØN DELETE(K) IF(ID(CØNT(K))=2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)</pre>	
<pre>PONCTION DELETERN IF(ID(CØNT(K))-2)1,2,1 2 PRINT 100 100 FØRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.) DELETE=0. RETURN 1 DELETE=CØNT(K+1)</pre>	
<ul> <li>PRINT 100</li> <li>PRMAT(40HATTEMPT TØ DELETE HEADER. ZERØ RETURNED.)</li> <li>DELETE=0.</li> <li>RETURN</li> <li>1 DELETE=CØNT(K+1)</li> </ul>	
<pre></pre>	
DELETE=0. RETURN 1 DELETE=CØNT(K+1)	
RETURN 1 DELETE=CØNT(K+1)	
1 DELETE=CØNT(K+1)	
LL=LNKL(CØNT(K))	
LR=LNKR(CØNT(K))	
CALL REELL(K)	
CALL SETIND(-1,-1,-L)	
ETORN END	
END CINE	
•	
0	
	1.000
0	

-	
	ZZEØR
0	LET DI
0	77 CEORDP 0607
	ZZ SEGRUR UG91
	KEANDKAA1A
ω	*FANDK0810
	*LDISKSEQRDR0697
	FUNCTION SEQRDR(LST)
	SEQRDR=CØNT(LØCT(LST))
-	RETURN
•	END
10	
60 0	
-	
A N	
ž	
510	
-	
R-U	
E E	
•	
-	
-	
11	
1	
-	
- 4	
2	
•	
-	

•	
de la	
1-12-	77 SEQUE 0699
	ZZ SEGER 0077
â	ZZFØR
	*FANDK0810
0	*LDISKSEQLR 0699
	FUNCTION SEQLR(Z,N)
	L = LNKR(Z) $Z = CONT(L)$
0	SEQLR=CONT(1+1)
	N = ID(Z) - I
	RETURN
	END
• °	
ž	
08	
0CK	
5	
~	
Į	
- Lor	
NU-	
IN COLUMN	
OS	
0	
0	
11	
io	
0,	
6	
5	
<b>U</b> a	
• • ²	
1	

- Tim	
2	ZZ SEQLL 0701
0	7.7E@D
20	ZZFØR
0	*EANDKORIO
so .	* ANDRUGIU
	EUNCTION SECULIZIAN
	LINKL(7)
-	
	SEVEL=CONT(L+1)
	N=ID(Z)=1
	KETUKN
	END
3	
×	
L C	
ER.	
•	
10	
19	
-	
-	

-	
+ <b>♀</b> .	77 SEQSR 0703
0	ZZFØR
	*FANDK0810
0	*LDISKSEQSR 0703
	FUNCTION SEQSR(Z,N)
	$\frac{1}{1} = 1 \times \mathbb{R} \left( \frac{1}{2} - 1 \right) + \frac{1}{2} + \frac{1}{2} \times \mathbb{R} \left( \frac{1}{2} + \frac{1}{2} + \frac{1}{2} \right) + \frac{1}{2} + \frac{1}$
0	GØ TØ 3
	4 L=LNKR(Z)
0	3 IF(ID(CØNT(L))-1)1,2,1
8 -	$\frac{1}{7 = CONT(L)}$
05	N = ID(Z) - I
-	RETURN
2	2 L=LNKR(CØNT(L+1)))
0 ę_	END
Toci-	
0	
- Maria	
- INIT	
0.0	
- 1.5	
0	
0	
12	
O 10 _	
9 <u>-</u>	
0	
6	
5	
- A	
2	
0	

	77 05001 0705
	ZZ SEQSL 0705
	ZZFØR
	*FANDK0810
	*LDISKSEQSL 0705
	FUNCTION SEQSL(Z,N)
	$\frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}$
	GØ TØ 3
14 Z. 2	4 L=LNKL(Z)
	3 IF(ID(CØNT(L))-1)1,2,1
	1 SEQSL=CØNT(L+1)
	Z=CØNT(L)
	N = ID(Z) = I
	$\frac{RETURN}{2 I - I NKL(CONT(I+1))}$
	$\frac{60}{10}$
	END
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	<pre>\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.</pre>
	SDEOF DEFINE END OF FORTRAN PROGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	\$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.
	SDEØF DEFINE END ØF FØRTRAN PRØGRAM FILE R.L.S.

•	
-0-	
	ZZJØB 29270126507387+215
	ZZJØB 29270126507387+215
-	
	77 KPUT 0694
0	77606
	LLSPS
178 - S 12	
0	*LIST CARD
	*ASSEMBLE RELØCATABLE
	*STØRE RELØADABLE
1	*NAME KPUT
0.	*ID NUMBER 0694
	***** KPUT 0694
0	
3.401	
-	DAC 6, KPUT , /-Q
08	DVLC22-Q,5,LENGIH,2,8,2,10,5,ENI-6,5,0,20,0
	DSC 17,0,0
0 -	DØRGQ-100
n -	EAC DS •2492
0	EX7 DS 3099
O H	IPNI DS 5
INN	ICHAR DSA
÷ -	WØRD DSA
- CBI	LST DSA
08-	DC 1.
	De 5
0	
	AM 1F+0,4,10
	AM ENT-1,5,10
0	TF CF+11,1-ENT
	BNF *+36,CF+11
12 -	CF CF CF+11
1.1_	TF CF+11,-CF-11
010	TE TE •CE+11
8	
0	DINK ENTIL2 - IF-0
	AM ENI-1,2,10
- e	BSBA*+12
5-	CM -WØRD,,10
	BNE *+24
з.	TEM IPNT
2	

.

	AM LENT 2 10
	CM IPNT-12-10
	BNE NOT
	BIM **+11
	NEWBRIDS *=5
	DSA -WARD
	DSA HIST
	TE -WØRD EXZ
	TEM IPNT-2-10
	NOT BLX $*+12$ , IPNT(A1)
A Street of the	TE FACWØRD
	TE BETA - ICHAR
	TD FAC-11(A1),BETA-1
	TD FAC-10(A1),BETA
	TE -WØRD.FAC
	BSBB1-ENT
	QQ DAC 6, NEWBØT,
	DVLC, 5, NEWBØT
	DC 2,'
	LENGTHDS ,QQ-1
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND SDEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND SDEØF DEFINE END ØF S.P.S. PRØGRAM FILE R.L.S.
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE == R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+
	DEND \$DEØF DEFINE END ØF S+P+S+ PRØGRAM FILE R+L+S+

-0 ZZFØRX A *FANDK0810 10 CALL LISTRD(LIST(K),5) CALL LDUMP(1,99) CALL IRALST(K) GØ TØ 10 END XXX (1 2 3 4 555 6 7.0 8 0.9 10 )ZZZZZZZZ (123.456 +123.456 -123.456 123+456 123-456 ) (ABC DEF GHIJ ABCDEFGHIJ \$ * GØ.NØW ) (INHALT=CØNT(LIST, ARG). DØN'T YØU THINKSØ+ NØ ) (A-B=A+B. A+B=A-B. A+3 NØT= A +3 . NEITHER A+3. END) ((A/B A,B (SUB3S , TEX5T, NUM 3) Z)

	ZZZZ END ØF JØB
-	
17211	
The strength of the	
0 <u> </u>	
X	
-	
•	
3	



# THE ENGLISH ELECTRIC COMPANY LIMITED

## Nelson Research Laboratories

TELEGRAMS : ENELECTICO STAFFORD TELEPHONE : STAFFORD 3271 BEACONHILL STAFFORD

4th October, 1967.

Prof. J. Weizenbaum, Project MAC, MIT, 545 Technology Sq., Cambridge, Mass. U.S.A.

Dear Sir,

I would be grateful if you could send me more information about SLIP. Could you please inform me how I might obtain a programming manual for this language?

Yours faithfully,

Gerald &. Edwards.

G.G. Edwards Head of System Programming

HEP

# NEWARK COLLEGE OF ENGINEERING

323 HIGH STREET

NEWARK, N. J. 07102

COMPUTING CENTER

AREA CODE 201, 624-2424

RECEIVED PROJECT MAC OCT 2 9 1965 1965

Dr. Joseph Weizenbaum M.I.T. Project MAC Cambridge, Mass.

Dear Dr. Weizenbaum:

By way of response to your letter in the Communications (ACM, May 1965), I'm writing to inquire whether you know of any implementations of SLIP for any IBM 1620 FORTRAN system -- or, indeed, if you know of any insurmountable difficulties in that direction.

I really don't inted this as a firm declaration of intent because we aren't sure we can get the time or the help, but we would like to.

Sincerely yours,

Dr. Phyllis Fox Associate Director

c.c. Mr. Hub Seward PF:hv

# THE UNIVERSITY OF BRITISH COLUMBIA

VANCOUVER 8, CANADA

COMPUTING CENTRE

December 23rd, 1965.

Professor J. Weizenbaum, Electrical Engineering, Massachusetts Institute of Technology, Cambridge 39, Massachusetts. 02139.



Dear Professor Weizenbaum:

Thank you for your letter of October 8th, suggesting where I might obtain a 7040 version of SLIP. I have now received such a system on magnetic tape from the Computer Center at the University of Pennsylvania. My original contact was with Professor John W. Carr at the Moore School, but the tape and related correspondence were sent by Mr. Paul Wolfgang at the Computer Center. Mr.Wolfgang also informs me that they are preparing a manual on the use of SLIP.

Perhaps this information will be of use to you.

Yours sincerely,

J.R.H.Dempster.

JRHD/1s.

## ASSOCIATION FOR COMPUTING MACHINERY

#### 211 EAST 43 STREET NEW YORK 17, NEW YORK

February 10th, 1965.

Mr. Donald B. Russell National Institute for Research in Nuclear Science Atlas Computer Laboratory Chilton, Didcot, Berkshire England

Dear Mr. Russell:

Thank you for your letter on SLIP. I am forwarding it for publication in the Communications of the ACM.

Yours truly,

lly

C. C. Gotlieb Editor-in-Chief Comm. ACM

CCG:mc

cc: Miss M. R. Kellington Prof. J. Weizenbaum

P.S. to Prof. Weizenbaum:

Russell's letter only came yesterday is via surface mail because the stamp must have fallen off. I hate to say I told you so but this is why I argued so hard, but unsuccessfully, with you and Tom Cheatham for omitting the program listings on SLIP. Not withstanding, the SLIP paper was an excellent contribution.

## UNIVERSITY OF MARYLAND

COLLEGE PARK 20742

COMPUTER SCIENCE CENTER

August 19, 1965

RECEIVED PROJECT MAC AUE 2 3 1965

Professor J. Weizenbaum Project MAC Massachusetts Institute of Technology Cambridge, Mass.

Dear Professor Weizenbaum:

In order to encourage the use of SLIP at our Center several staff members familiar with the package have produced some documentation intended for use by programmers whose lack of familiarity with list processing inhibits their learning the system through the technical publications.

The work done to date would seem to form a good basis for an introduction to list processing based on SLIP. Before investing further efforts along these lines, we would like to insure that we are not duplicating efforts which are being made elsewhere.

We are writing to you to determine whether or not you know of work being done to produce an introductory "user's manual" for SLIP. For your convenience, we are enclosing one of our reply cards which you may use to indicate any projects that you may be **a**ware of.

We will greatly appreciate any help that you can give us in this matter. Thank you for your assistance.

Very truly yours,

David L. Parnas Assistant Professor

srw enc.

#### NORTH AMERICAN AVIATION, INC. M. A. Holly D/196-210, EA43



SPACE and INFORMATION SYSTEMS DIVISION

#### NORTH AMERICAN AVIATION, INC.



January 11, 1966

in reply refer to:



Mr. J. Weizenbaum Computer Laboratory General Electric Company Sunnyvale, California

Dear Sir:

I am currently interested in using your SLIP programs on the 7040-44. I would appreciate any information you might have concerning the availability of SLIP decks for this machine.

Sincerely yours,

NORTH AMERICAN AVIATION, INC.

Mike Holly

M. A. Holly Advanced Applications Information Systems

sb

# Leicester Colleges of Art and Technology LEICESTER, ENGLAND REGISTRAR: A. MARTIN, F.C.1.S.

10th March 1966 PGR/BS

Prof. J.Weizenbaum, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass, U.S.A. RECEIVED PROJECT MAC MAR 1 5 1966

Dear Sir,

Could you please let me have an up-to-date listing of the SLIP system.

Yours faithfully,

P.G. Raymont.

(P.G.Raymont) Director of Computing Laboratory.





UNIVERSITY OF MARYLAND COLLEGE PARK 20742

COMPUTER SCIENCE CENTER

May 20, 1965

Dr. Joseph Weizenbaum M.I.T. Project MAC Cambridge, Mass.

Dear Dr. Weizenbaum:

In response to your request in the Communications of the ACM (May 1965), I would like to briefly give some information on our implementation of your SLIP system.

If you recall, Mr. John L. Pfaltz wrote you a letter in December of 1964 stating our intention to use SLIP. Since then, I have worked with Mr. Pfaltz in preparing most of the programs listed in the Communications of the ACM (September 1963). The Computer Science Center here at the University of Maryland, has an IBM 7094/1401 complex. The programs for the 7094 have been written in FORTRAN IV. In addition to the errors mentioned in the Communications of the ACM (May 1965), we have found for the function NAMEDL that the instruction should read

NAMEDL = LNKL(CONT(LOCT(L)+1)).

Furthermore, the function DELETE fails to decrement the reference counter of the sublist when the cell deleted is a "name" cell.

Some cautions might also be mentioned as potential mistakes. When testing a cell by NAMTST, only the double address convention and the "supposed" header is checked to insure that the cell is referring to a "name". For PRLSTS and in similar situations, if one uses the list alias as a piece of datum (ID=0), PRLSTS will consider it as if the ID=1. This necessitates a simple test of the ID portion of the cell in PRLSTS (note, not in NAMTST).

Modifications and additions to your system have been written; we incorporated, for example, variable spacing (to indicate sublevels) in PRLSTS and a new - 2 - Lieberman/Weizenbaum

program to print a list "linearly". Also being developed are systems to store and process general tree-like structures using SLIP. To further simplify and to extend the capabilities of SLIP, other routines are being written. By the summer, we hope to have a paper describing SLIP for users at the University of Maryland. This will include technical and non-technical sections.

We would be most interested in other groups using your system and, therefore, are looking forward to such a list appearing in the Communications of the ACM as you suggested.

Thank you for your time and interest.

Sincerely yours,

Robert M. Lieberman

Robert N. Lieberman Research Graduate Assistant

RNL:ags

211 EAST 43 STREET NEW YORK 17, NEW YORK

C. C. GOTLIEB, Editor-in-Chief MYRTLE R. KELLINGTON, Executive Editor LIN S. WILLIAMS, Advertising Manager

COMMUNICATIONS OF THE ACM A Publication of the Association for Computing Machinery



**REPLY TO:** INSTITUTE OF COMPUTER SCIENCE McLennan Laboratory University of Toronto Toronto 5, Canada

February 8th, 1965.

Prof. Joseph Weizenbaum M.I.T. Project MAC Room 809 545 Technology Sqr. CAMBRIDGE 39, Mass.

212 YU 6-3055

Dear Joe:

I am forwarding your letter on SLIP for publication. I am also including a note from Robert Novak who has been working with SLIP here. Dr. J.M. Kennedy of the Atomic Energy Company Limited, Chalk River, Ontario, mentioned to me some time ago that he too had implemented SLIP on their Bendix G20 computer. It might be worth your while bringing your letter to his attention and inviting a submission from him too.

Best regards.

lalm 9. Thit

C. C. Gotlieb, Editor-in-Chief, Comm. ACM

CCG/ez

c.c. M.R. Kellington

SLIP System at Institute of Computer Science, University of Toronto

R. Novak

SLIP at the University of Toronto for the I.B.M. 7094/II is a primitive based system. The primitive functions (ID, LNKL, LNKR, CØNT, INHALT, MADØU, SETDIR, SETIND, STRDIR, STRIND) were written in MAP assembly language for the 7090. (They have not been changed for the 7094.) Most other functions are FORTRAN IV versions of those listed in the Communications (C.A.C.M. September 1963 p.524).

Included are some routines from the SLIP version supplied by North American Aviation (i.e. the linear advance routines and some text manipulation routines). The only additions made were to include two primitive functions to make use of unused bits.

The routines that allow for recursive calling of functions are not implemented directly, but arrangements have been included to allow for recursive call of subroutines in the system for algebraic manipulation for which SLIP is being used.



## LEARNING RESEARCH AND DEVELOPMENT CENTER

UNIVERSITY OF PITTSBURGH•PITTSBURGH•PENNSYLVANIA 15213 MAIN OFFICE: 302 AMOS HALL•PHONE 621-3500 EXT. 7226-7227 LABORATORIES: UPPER CAMPUS•PHONE 683-1620 EXT. 2422-23-24

February 8, 1965

Dr. Josef Weisenbaum Mass. Institute of Technology Cambridge, Massachusetts

Dear Dr. Weisenbaum:

I am writing to inquire about the current status of SLIP and to ask advice in terms of implementing it on our computer. Rumors concerning the status of SLIP vary from "Weisenbaum had abandoned it" to somewhat more optimistic statements. I attended your course last year and I'm intrigued with SLIP's potential. I know, however, that several bugs were detected subsequent to publication and that you added, as I recall, several more functions. Do you have a listing of the corrections and additions that I might have?

In terms of the second matter, I have made only one abortive attempt while at Harvard to use SLIP; thus, I am not really familiar with the amount of core that the system requires, etc. Shortly, we will have a PDP 7 with at least 8K memory (18 bit words). We may augment this by the time the computer is delivered either by purchasing more core and/or a drum. In view of the obvious restrictions, e.g. limited command structure, 18 bit words, etc., is it feasible to try to implement SLIP on the PDP? I think that it would be a very useful tool to add to our soft ware.

Thank you for your consideration.

Sincerely,

Kjillergand Paul M. Kjeldergaard

Research Associate

/jnm



AERO - SPACE DIVISION + P.O. BOX 3707 + SEATTLE, WASHINGTON 98124

September 24, 1964 IN REPLY REFER TO 2-5305-3-783

Professor J. Weizenbaum MIT Computation Center Cambridge, Massachusetts

Subject: SLIP List Processing Language

Dear Professor Weizenbaum:

Mr. Michael R. Quamme of our Engineering Data Systems Group has asked us to coordinate his request for the subject program. The availability of the SLIP Program was confirmed in a telephone conversation, September 16, between Mr. Quamme and your Mr. D. G. Bobrow. We are sending you a blank tape for this program transmittal.

We would also appreciate any supporting written information in the following two areas:

- 1. Program documentation and program teaching aids.
- Requirements for program implementation on a three channel, eighteen tape unit IBM 7094 having IBSYS 9 available as executive monitor.

Please direct the tape and any further correspondence to the Applied Mathematics Information Center, Organization 2-5305, Mail Stop 22-75, 2.01-2 Building, Bay L-9.

Yours truly,

THE BOEING COMPANY Aero-Space Division

(mr.) Muray Tumakura

fn W. S. Lagen Applied Mathematics Information Center

North Building B017 Indiana University Bloomington, Indiana 29 June 1964

Dear Joe,

At present I am doing research at the U.of Indiana with a Language Data Processing seminar sponsored by the Air Force. A number of people here are interested in programming languages for linguistic problems and through my encouragement have obtained an interest in SLIP. (The people here are faculty members from about 12 different universities who are doing or are planning to do computer-aided research.) Do you have any additional literature on SLIP other than the G.E. manual which you gave me about a year ago and the ACM article which you wrote?

Sandy Elkin mentioned a few months ago that you had coded a SLIP system for processing COMIT-like statements.( I don't recall if he said he heard this from you or from someone else.) If you have done this, could I possibly obtain from you a listing of the system? Also, can 709 users obtain SLIP source decks from SHARE? Several people here may want to try SLIP on Indiana's 709.

Besides, these questions, I would be interested in hearing about your programming work in general, if you have the time to describe it. At present I am working on spelling-to-sound relationships in English. The work is geared towards a reading machine for the blind and towards the improvement of the teaching of reading. At the end of August I will return to Stanford where **x** I have about 6 more months of work for completing my Ph.D. in linguistics.

Sincerely,

DickV

Dick Venezky

LITTON INDUSTRIES GUIDANCE AND CONTROL SYSTEMS DIVISION 5500 CANOGA AVENUE • WOODLAND HILLS, CALIFORNIA 91364 346-4040 • CABLE ADDRESS LITTIND

June 5, 1964

Mr. Joseph Weizenbaum General Electric Computer Laboratory P. O. Box 1285 Sunnyvale, California

Dear Mr. Weizenbaum:

I am writing you in regard to SLIP which appears in Communications of the ACM, September, 1963.

The subroutines TERM, VISIT, and INT/GER are not easily understood by me in terms of return linkage of the function subroutines. Our compiler generates a load card sequence at the end of FUNCTION LSTEQL card #551. In card #667 VISIT is a subroutine with a single valued argument in contrast to a function with 2 values in card #531.

Could you help me clear this up, as we are eager to use the Slip system? A listing of the three subroutines (functions) should be adequate if they are I.B.M. oriented. Would you send any information you collect to my home address:

> R. A. White 20509 Schoenborn Street Canoga Park, California

> > Sincerely,

Richard G. Maite

Richard A. White

RAW:nam
Résidence : 731-6004.

Bureau: 737-0862.



Les Frères de Saint-Gabriel

ADMINISTRATION PROVINCIALE 5638, rue Canterbury Montréal 26.

November 29, 1964

PROVINCE de MONTRÉAL

Professor J. Weizenbaum M.I.T. Cambridge, Mass., U.S.A.

Professor,

Would you be so kind as to let me know where I could get full information about the SLIP programming language, that could lead to a complete training in the use of that intermediate language for the electronic treatment of linguistic problems.

Early in 1965, the University of Montreal will be equipped with a CD 3400. The Department of Linguistics intends to make use of the facilities offered by the Computational Center for intense research in Automatic Translation and various types of Linguistic Structures.

I am told that the SLIP programming language will be used by the Department of Linguistics for such purposes. I would greatly appreciate receiving full details about the literature available on SLIP.

Yours sincerely,

binist Fanbert

Ernest Faubert, professor.

## UNIVERSITY OF COLORADO

## BOULDER. COLORADO

GRADUATE SCHOOL

COMPUTING CENTER

September 28, 1964

Joseph Weizebaum General Electric Computer Laboratory P. O. Box 1285 Sunnyvale, California

Dear Mr. Weizenbaum:

After reading the Sept. 63 issues of the communications of the ACM I became interested in SLIP. When I started to implement it on the Universities 709 I found that several routines were not included in the appendix to the above article. Was this intentional or has a page(s) been omitted from the appendix? The routines I need, unless there have been some keypunching errors, are: SQIN, SQOUT, CRDBU, EQUAL, VISIT, LANORM, SHIN, TERM, INTEGER.

Any additional information on SLIP would be most welcome.

Sincerely

Pairl E Schultz

David Schultz Programming Advisor

CARNEGIE INSTITUTE OF TECHNOLOGY SCHENLEY PARK PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION TELEPHONE: 621-2600 AREA CODE 412

June 16, 1964

Mr. Joseph Weizenbaum General Electric Computer Laboratory P.O. Box 1285 Sunnyvale, California

Dear Mr. Weizenbaum:

I am very much interested in using SLIP in developing a model of the list-processing structure of the New York Stock Exchange, and would be most appreciative if you could send me any further documentation on the system beyond that contained in your September 1963 article in the ACM <u>Communications</u>. I would also be most appreciative if I could get a source program deck for the FORTRAN/FAP version of the system so as to avoid using the slower all-FORTRAN version reprinted in the appendix to your article and having to program primitives. I will be using the system on a 7090 at the University of Pittsburgh -or rather, will be through Ted Van Wormer, who will help in any modifications which might be required.

Thank you.

Sincerely,

Ilan bann

John Bossons

JB/jlb

## SUN OIL COMPANY

C. G. KIRKBRIDE VICE PRESIDENT · RESEARCH & ENGINEERING 1608 WALNUT STREET PHILADELPHIA 3,PA.

November 18, 1964

W. E. BONNET DIRECTOR - TECHNICAL ECONOMICS RESEARCH & ENGINEERING

Mr. Joseph Weizenbaum General Electric Computer Laboratory P. O. Box 1285 Sunnyvale, California

Dear Sir:

Would it be possible to get a source deck of your SLIP language? We are currently developing a general information system which will require a list processing language. SLIP looks particularly interesting because of its Fortran compatibility and the promising speed advantages from symmetry. With the deck as a starting point, we can investigate its potential, what modifications might be required, and so on.

Thank you.

Very truly yours,

40.

Bamberger

David R. Bamberger

1ap

## COPY

November 5, 1963

Mr. R. W. Elliott 1103-A Brackenridge Apts. Austin 3, Texas, 78703

Dear Mr. Elliott:

I am currently at M.I.T. and will remain here for the rest of the academic year. My moving about accounts for the loss of your earlier letter. I do apologize for the poor service.

Since I have access to only a 7094 here-i.e. not a 1604-I am in a very poor position to send out copies of the SLIP deck for the 1604. I have arranged to have a copy of the deck you want in the hands of Dr. Sandy Elkin at Control Data in Palo Alto. The whole system should soon become part of the COOP library in any case. For the present, I suggest you write to Sandy and get him to send you a tape or a copy of the deck.

Another drawback is that I have added to the system in the 7090/7094 context. My current system is therefore somewhat more powerful than the published version. I plan to integrate the whole thing one day by means of having a SLIP meeting (probably on the East Coast) to which all interested individuals will be invited. I mention this now in order to get your reaction to this idea and to solicit from you a statement as to the optimum time and place for such an event. The question on the length of such a meeting is also open.

Sincerely yours.

J. Weizenbaum Visiting Assoc. Prof. M.I.T. - Project MAC

JW/jep

1103-A Brackenridge Apts. Austin 3, Texas 78703 October 29, 1963

Dr. Joseph Weizenbaum General Electric Computer Laboratory P. O. Box 1285 Sunnyvale, California

Dear Dr. Weizenbaum:

I wrote to you about a month ago but have received no reply so I will assume that my previous letter was lost somewhere along the way.

We here at the University of Texas are interested in implementing your SLIP system on the CDC 1604 in our computation center. It is my understanding that you have already implemented a SLIP system on a 1604. If possible we would like to obtain a punched deck from you.

In addition, any additional information, especially a programming manual, that you could provide would be appreciated. We have funds to purchase any items that are available.

Any reply, either to myself or to Dr. R. K. Lindsay, Computation Center, University of Texas, will be appreciated.

Thank you.

Very truly yours,

ogen W. Ellist

Roger W. Elliott