1963-67 OK

SLIP 1 of 2

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

Dear Editor:

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) time-sharing 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.

JOSEPH WEIZENBAUM
M. I. T. Project MAC
Cambridge, Mass.

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 up to date in the system . As of now SLIP is on the Library tape on the 1604 and is being used (and taught) in two computer science courses . Next semester I will attempt to write a general manual on SLIP to supplement the documentation in the Communications . If any work has been done in this vein I would appreciate information about said . To 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 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 LSSCPY ---- after line 559 add

```
    if( LNKR(W) - LPNTR(LR)) 6, 7, 6
7   X = ADVLWR( LR, K)
    GO TO 5
6   IF( LNKR(W(20 - LPNTR( LR)) 5. 8. 5
8   X = ADVLWR ( LR, K )
```

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
41  IF( LNKR ( W) - LPNTR( LRA) ) 11, 12, 11
12  XA  = ADVLWR ( LRA, KA )
11   IF ( LNKR ( W ) - LPNTR( LRB)) 42, 13, 42
42   if( LNKR(W(2) - LPNTR ( LRB )) 8,13, 8
13   XB = ADVLWR ( LRB, KB )
```

3. Function LSSCPY ---- After line 562 add

```
     L = LNKR ( CONT ( W(2)))
     CALL STRIND ( 0, L+ 1 )
     Call SETIND ( 0,-1,-1, L )
```

This 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 and has been deleted comes up to the top of LAVS .

4. Function LVLRV1 ----- line 434 should read

```
IF( LNKR( CONT (  LVLRV1  + 1 )) 2, 3, 2
```

5. function LVLRVT ----- Line 423 should read

```
1     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 Dept. UNiv of Wisc. the faculty member under whom I am working as I shall not be here over the summer .

Yours truly

Marvin Tatner

Marvin Tatner

P.S. errors continued

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

```
IF ( LISTMT ( L )) 2.1.2
```

This corrects 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 W(2) . This however seems more in line with normal list creations by the programmer .

MAY 21, 1966

LISTING OF 1620 SLIP SYSTEM

DICK SITES
DAVID P. KELLEHER

```
ZZ    LNKL   0630

ZZSPS
*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME LNKL
*ID NUMBER 0630
A1000****  LNKL   0630
A1010Q     DS   ,*+101
A1020      DC   6,987898,5-Q
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
A1060      DØRGQ-100
A1070FAC   DS   ,2492
A1080BETA  DS   ,2630
A1090FXZ   DS   ,3099
A1100      DS   5
A1110ENT   AM   *-1,5,10
A1120      TF   CF+11,1-ENT
A1130      BNF  TF,CF+11
A1140CF    CF   CF+11
A1150      TF   CF+11,-CF-11
A1160TF    AM   ENT-1,2,10
A1170      TFL  BETA,-CF-11
A1180      TF   FAC,FXZ
A1190      MA   FAC,BETA-5
A1200      B7   1-ENT
A1210LENGTHDC   2,'
A1220      DEND
```

```
ZZ    LNKR   0632

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME LNKR
*ID NUMBER 0632
B1000****  LNKR   0632
B1010Q     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
B1090      DS   5
B1100ENT   AM   *-1,5,10
B1110      TF   CF+11,1-ENT
B1120      BNF  TF,CF+11
B1130CF    CF   CF+11
B1140      TF   CF+11,-CF-11
B1150TF    AM   ENT-1,2,10
B1160      TF   FAC,FXZ
B1170      MA   FAC,-CF-11
B1180      B7   1-ENT
B1190LENGTHDC   2,'
B1200      DEND
```

```
ZZ    ID     0634

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME ID
*ID NUMBER 0634
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
C1070FAC   DS   ,2492
C1080BETA  DS   ,2630
C1090FXZ   DS   ,3099
C1100      DS   5
C1110ENT   AM   *-1,5,10
C1120      TF   CF+11,1-ENT
C1130      BNF  TF,CF+11
C1140CF    CF   CF+11
C1150      TF   CF+11,-CF-11
C1160TF    AM   ENT-1,2,10
C1170      TFL  BETA,-CF-11
C1180      TF   FAC,FXZ
C1190      BNF  *+24,BETA-1
C1192      AM   FAC,1,10
C1194      BNF  *+24,BETA-2
C1196      AM   FAC,2,10
C1200      B7   1-ENT
C1210LENGTHDC   2,'
C1220      DEND
```

```
ZZ    CØNT   0636

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME CØNT
*ID NUMBER 0636
D1000****  CØNT   0636
D1010Q     DS   ,*+101
D1020      DC   6,987898,5-Q
D1030      DAC  6,CØNT   ,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
D1080FXZ   DS   ,3099
D1082CALL  BTM  ,*+11
D1084INITASDS   ,*-5
D1090      DS   5
D1100ENT   B    CALL,,,TRANSFER VECTØR
D1105SLADR DS   ,*-5
D1109      AM   ENT-1,5,10
D1110      TF   CF+11,1-ENT
D1120      BNF  TF,CF+11
D1130CF    CF   CF+11
D1140      TF   CF+11,-CF-11
D1150TF    AM   ENT-1,2,10
D1160      MA   *+35,-CF-11
D1165      MF   *+23,-CF-11
D1170      BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
D1175      TFL  FAC,-99
D1180      B7   1-ENT
D1190QQ    DAC  6,INITAS,
D1192      DVLC,5,INITAS
D1194      DC   2,'
D1196LENGTHDS   ,QQ-1
D1200      DEND
```

```
ZZ    INITAS 0640

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME INITAS
*ID NUMBER 0640
E1000****  INITAS 0640
E1010Q     DS  ,*+101
E1020      DC  6,987898,5-Q
E1030      DAC 6,INITAS,7-Q
E1040      DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
E1050      DSC 17,0,0
E1053      DØRGQ-100
E1057HIGH  DS  ,434
E1061CØMADDDS  ,2231
E1062FAC   DS  ,2492
E1063BETA  DS  ,2630
E1064FXZ   DS  ,3099
E1066FX1   DS  ,3109
E1070      DS  5
E1080ENT   AM  *-1,7,10
E1082      B7  *+8
E1084      TFM *-2,NEXT
E1090****  LINK TRANSFER VECTØRS TØ EXTERNALLY USED UTILITY ENTRY PØINTS
E1100      TRNM,SLADR.
E1110CØNT  DS  ,*-5
E1111      TRNM,SLADR.
E1112SETINDDS  ,*-5
E1113      TRNM,SLADR.
E1114STRINDDS  ,*-5
E1220****  CHAIN UP FREE SPACE - INITIALIZATIØN ØF CHAINER
E1230      TF  LWRLM,HIGH
E1235      TF  UPRLM,CØMADD
           BNC3*+24
           TFM LWRLM,57986
E1240      TDM LWRLM,9
E1260      S   UPRLM,LWRLM
E1261      CM  UPRLM-2,,10
E1262      BNE *+32
E1263      TFM MES,NMS2
```

```
E1264       B7  ERRØR
E1265       TF  DISK+8,UPRLM-2
E1267       TDM UPRLM-1
E1270       TDM UPRLM
E1280       A   UPRLM,LWRLM
E1283       AM  LWRLM,1,10
E1286       TF  DISK+13,LWRLM
E1289       AM  LWRLM,19,10
E1292       TF  AVSLL,AVSLR
E1294       SM  AVSLL,5,10
E1300****   PERFØRM CHAINING - MAY BE RE-USED BY ENTERING AT XXX
E1301****       AFTER SETTING BLKN AND RETN+6.
E1310XXX    BTAMSLADR,,78
E1312BLKN   DS  ,*-4
E1315       TF  FAC,FX1
E1316       TD  FAC-4,BLKN
E1317       CF  FAC-4
E1318       TF  -AVSLR,FAC
E1319       MA  -AVSLL,-AVSLR
E1320       TD  MES1+12,BLKN
E1330       WATYMES1
E1380LØØP   AM  99,10,10
E1390       AM  FAC,2,10
E1400       TF  -99,FAC
E1410       AM  99,10,10
E1430       TF  -99,FXZ
E1440       AM  -AVSLL,2,10
E1450       C   99,UPRLM
E1460       BNE LØØP
E1462       SM  99,10,10
E1464       TF  -99,FXZ
E1470       TF  BETA,-AVSLL
E1472       SF  BETA-3,,7
E1473TØTAL  DS  ,*
E1474       MM  BETA,5,10
E1476       A   TØTAL,98
E1480       TNF MES2+8,TØTAL
E1485       MA  FAC,TØTAL
E1490       WATYMES2
E1500       SM  -AVSLL,2,10
E1505RETN   B7  1-ENT
E1525NEXT   TF  FAC,FXZ
E1526****   ATTEMPT TØ LØCATE MØRE SPACE
E1999       B7  1-ENT
```

```
E2000****  EXTERNALLY CALLED UTILITIES
E2011*
E2012****  CALLING SEQUENCE = BTAM,SLADR,SLIP-ADDRESS,78 / RETURNS
E2013****      MACHINE ADDRESS IN 99 AFTER PØSSIBLY TURNING A DISK PAGE.
E2020      DAC 3,...
E2022SLADR TF  99,LWRLM
E2025      BNC3NØTRP
           PUT SWAP
E2031NØTRP BNF CHEK,SLADR-1
E2032      BB2
E2035CHEK  BNV ADD
E2036      CF  BACK+10
E2040ADD   A   98,SLADR-1
E2045      BV  ERR1
E2050      CM  SLADR-5,,1011
E2060PAGE  DS  ,*
E2070      BE  INCØR
E2080****  BLØCK NØT IN CØRE
E2290      H   1
E2400INCØR C   99,UPRLM
E2410      BH  ERR1
E2411BACK  AM  *+8,,7910,TURN PØSSIBLE CHECK BACK ØN
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)
E3001MES2  DMES,A,XXXXX SLIP CELLS INITIALIZED.(E)
E3010NMS1  DMES,A,PSEUDØ-ADDRESS XXXX ØUT ØF RANGE(E)
E3015NMS2  DMES,A,SLIP ØVERLAP(E)
E3120****  INTERNALLY REFERENCED CØNSTANTS
E3130LWRLM DS  5,,ADDRESS ØF RIGHTMØST DIGIT ØF PSEUDØ-WØRD NUMBER (X0000)
E3132UPRLM DS  5,,ADDRESS ØF  LAST PSEUDØ-WØRD (END ØF SECTØR BLØCK)
E3140DISK  DDA
E3145SWAP  DD  ,DISK
E3340AVSLL DS  5,    , PØINTERS TØ FIELD ADDRESSES ØF RIGHT AND LEFT
E3350AVSLR DS  ,LWRLM, LINKS ØF AVSL IN CØRE (SLIP CELL X0000).
E3440****  EXTERNALLY USED UTILITY ENTRY LINKAGES
E3450SLADR.NØP SLADR
E3460      DØRG*-4
```

```
E3470       DC  1,'
E3680QQ     DAC 6,CØNT  ,
E3690       DVLC,5,CØNT
E3691       DAC 6,SETIND,
E3692       DVLC,5,SETIND
E3694       DAC 6,STRIND
E3696       DVLC,5,STRIND
E3700       DC  2,'
E3710LENGTHDS  ,QQ-1
E3720       DEND
```

```
ZZ    SETDIR 0644

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME SETDIR
*ID NUMBER 0644
F1000****  SETDIR 0644
F1010Q     DS  ,*+101
F1020      DC  6,987898,5-Q
F1030      DAC 6,SETDIR,7-Q
F1040      DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
F1050      DSC 17,0,0
F1060      DØRGQ-100
F1065FAC   DS  ,2492
F1070ID    DSA
F1080LNKL  DSA
F1090LNKR  DSA
F1100CELL  DSA
F1110      DC  1,'
F1120      DS  5
F1130ENT   TFM TF+6,ID-4
F1140      AM  TF+6,4,10
F1150      AM  ENT-1,5,10
F1160      TF  CF+11,1-ENT
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
F2010      TFL FAC,-CELL
F2020      BNF SETID,-ID
F2030      B7  TEST2
F2040SETID MM  -ID,5,10
F2050      BD  *+32,99
F2060      CF  FAC-1
F2070      B7  *+20
F2080      SF  FAC-1
F2090      BD  *+32,98
```

```
F2100       CF   FAC-2
F2110       B7   *+20
F2120       SF   FAC-2
F2130TEST2  BNF  SETL,-LNKL
F2135       B7   TEST3
F2140SETL   MA   FAC-5,-LNKL
F2150TEST3  BNF  SETR,-LNKR
F2155       B7   DØNE
F2160SETR   MA   FAC,-LNKR
F2170DØNE   TFL  -CELL,FAC
F2180       B7   1-ENT
F2190LENGTHDC    2,'
F2200       DEND
```

```
ZZ    SETIND 0646

ZZSPS

*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
G1050      DSC  17,0,0
G1060      DØRGQ-100
G1070ID    DSA
G1080LNKL  DSA
G1090LNKR  DSA
G1100CELL  DSA
G1110      DC   1,'
G1112CALL  BTM  ,*+11
G1114INITASDS   ,*-5
G1120      DS   5
G1122ENT   B    CALL,,,TRANSFER VECTØR
G1123SLADR DS   ,*-5
G1130      TFM  TF+6,ID-4
G1140      AM   TF+6,4,10
G1150      AM   ENT-1,5,10
G1160      TF   CF+11,1-ENT
G1170      BNF  *+36,CF+11
G1180CF    CF   CF+11
G1190      TF   CF+11,-CF-11
G1200TF    TF   ,CF+11
G1210      AM   TF+6,1,10
G1220      BNR  ENT+24,-TF-6
G1230      AM   ENT-1,1,10
G2010      MA   *+35,-CELL
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 SUBPRØGRAM
G2050SETDIRDS   ,*-5
```

```
G2060          DSA -ID
G2070          DSA -LNKL
G2080          DSA -LNKR
G2090CELL.     DSA
G2180          B7  1-ENT
G2190QQ        DAC 6,SETDIR,
G2200          DVLC,5,SETDIR
G2202          DAC 6,INITAS,
G2204          DVLC,5,INITAS
G2210          DC  2,'
G2220LENGTHDS  ,QQ-1
G2230          DEND
```

```
ZZ    STRIND 0648

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*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
H1070DATUM DSA
H1080ICL   DSA
H1110      DC  1,'
H1112CALL  BTM ,*+11
H1114INITASDS  ,*-5
H1120      DS  5
H1125ENT   B   CALL,,,TRANSFER VECTØR
H1126SLADR DS  ,*-5
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
H1180CF    CF  CF+11
H1190      TF  CF+11,-CF-11
H1200TF    TF  ,CF+11
H1210      AM  TF+6,1,10
H1220      BNR ENT+24,-TF-6
H1230      AM  ENT-1,1,10
H2010      MA  *+35,-ICL
H2011      MF  *+23,-ICL
H2020      BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
H2030      TFL FAC,-DATUM
H2040      TFL -99,FAC
H2180      B7  1-ENT
H2190QQ    DAC 6,INITAS,
```

```
H2192       DVLC,5,INITAS
H2194       DC  2,'
H2196LENGTHDS  ,QQ-1
H2200       DEND
```

```
ZZ    KGET   0672

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME KGET
*ID NUMBER 0672
I1000****  KGET   0672
I1010Q     DS   ,*+101
I1020      DC   6,987898,5-Q
I1030      DAC  6,KGET  ,7-Q
I1040      DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
I1050      DSC  17,0,0
I1060      DØRGQ-100
I1061FAC   DS   ,2492
I1063BETA  DS   ,2630
I1064STZERØDS   ,2771
I1065FXZ   DS   ,3099
I1066****   =9, .=5, )=6, +=3, $=7, *=8, -=2, /=11, ,=13, (=10, ==1, '=14.
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+174
I1070INDV  DS   2,, 06 ØR 10
I1072      DGM
I1074NEXT  DS   ,PØINT-4, DTA ØR DCA
I1075MES   DMES,A, EXIT CALLED BY KGET(E)
I1076BUF   DAS  89
I1077      DC   1,',BUF+176
I1078INPT  DSA
I1080NCØDE DSA
I1110      DC   1,'
I1120      DS   5
I1130ENT   TFM  TF+6,INPT-4
I1140      AM   TF+6,4,10
I1150      AM   ENT-1,5,10
I1160      TF   CF+11,1-ENT
I1170      BNF  *+36,CF+11
I1180CF    CF   CF+11
I1190      TF   CF+11,-CF-11
I1200TF    TF   ,CF+11
I1210      AM   TF+6,1,10
```

```
I1220          BNR ENT+12,-TF-6
I1230          AM  ENT-1,1,10
I1240          BSBA*+12
I1300          TF  FAC,FXZ
I1310          TF  BETA,FXZ
I1320          BNF REG,-INPT
I1330          TR  -PØINT,RKM-2
I1340          B7  ØUT
I1345REG       AM  PØINT,2,10
I1350          BNR NØRED,-PØINT
I1380          TFM PØINT,BUF
I1390          MA  DEVIC,-INPT
I1400          A   DEVIC,DEVIC
I1410          BD  TYPE,DEVIC
I1420          TFM INDV,10,10
I1430          B7  *+20
I1440TYPE      TFM INDV,06,10
I1443****      CLEAR BUFFER -- IMAGE IS 87 INPUT CHARACTERS (BUF -- BUF+172),
I1444****           1 BLANK (BUF+174), AND 1 RECØRD MARK (BUF+176).
I1445RERE      TR  BUF-1,STZERØ-1
I1450          TR  BUF+88,STZERØ
I1460ALLBL     BD  *+24,INDV-1
I1462          RCTY
I1475          GET NEXT
I1476          BD  *+24,INDV-1
I1477          BC4 RERE
I148C          BLXM*+12,174(A1)
I148E          CM  BUF-2(A1),,10
I148G          BNE NBLK
I148J          BCXM*-24,-2(A1)
I148L          B7  ALLBL
I148NNBLK      TDM BUF+2(A1)
I148PRKM       DC  1,',*
I148R          BNR *+24,BUF-2(A1)
I148T          TFM BUF-2(A1),22,10
I148V          BCXM*-24,-2(A1)
I148W          BNC3NØRED
I149A          RCTY
I114C          WATYBUF
I114JNØRED     CM  -PØINT,22,10
I114L          BNE NØRKM
I1490          BNF ØKGØ,RMSW
I1492          WATYMES
I1500          CALLEXIT
```

```
I1525ØKGØ  SF   RMSW
I1530      B7   REG
I1540NØRKM TF   FAC,-PØINT
I1550      CM   FAC,70,10
I1560      BL   *+32
I1565****  NUMERIC = 12
I1570      TFM  BETA,12,10
I1580      B7   DØNE
I1590      CM   FAC,41,10
I1600      BL   *+32
I1605****  ALPHABETIC = 04
I1670      TFM  BETA,04,10
I1680      B7   DØNE
I1685****  NEITHER -- USE TABLE LØØK-UP
I1700      BLX  *+12,FAC(A1)
I1710      BX   *+12,FAC(A1)
I1720      TF   BETA,TABLE(A1)
I2010DØNE  CF   FAC-1,,10
I2020DEVIC DS   ,*
I2030RMSW  DS   ,*-2
I2040      CF   BETA-1
I2050      MF   BETA,RMSW
I2060      CF   RMSW
I20700UT   TF   -NCØDE,BETA
I2180      BSBB1-ENT
I2194LENGTHDC  1,'
I2200      DEND

$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE -- D.P.K.
```

```
ZZ    LCNTR  0637

ZZFØR

*FANDK0810
*LDISKLCNTR 0637
      FUNCTIØN LCNTR (LST)                                    LCNT0010
      LCNTR=LNKR (CØNT (LST+1) )                              LCNT0020
      RETURN                                                  LCNT0030
      END                                                     LCNT0040
```

```
ZZ    LSTNAM 0639

ZZFØR

*FANDK0810
*LDISKLSTNAM0639
      FUNCTIØN LSTNAM (LST)                                    LSTN0010
      LSTNAM=LNKL (CØNT (LST+1) )                              LSTN0020
      CALL SETDIR (0,LSTNAM,LSTNAM,LSTNAM)                     LSTN0030
      RETURN                                                   LSTN0040
      END                                                      LSTN0050
```

```
ZZ    INTLBL 0649

ZZFØR

*FANDK0810
*LDISKINTLBL0649
      FUNCTIØN INTLBL (I)                                    INTL0010
      INTLBL=I                                               INTL0020
      RETURN                                                 INTL0030
      END                                                    INTL0040
```

```
ZZ    FLTLBL 0651

ZZFØR

*FANDK0810
*LDISKFLTLBL0651
      FUNCTIØN FLTLBL (X)                                    FLTL0010
      FLTLBL=X                                               FLTL0020
      RETURN                                                 FLTL0030
      END                                                    FLTL0040
```

```
ZZ    NUCELL 0653

ZZFØR

*FANDK0810
*LDISKNUCELL0653
      FUNCTIØN NUCELL (X)                                          NUCE0010
      K=1                                                          NUCE0020
 10   M=LNKR(CØNT(-1))                                            NUCE0030
      IF(M)2,4,2                                                   NUCE0040
 4    GØ TØ (6,8),K                                                NUCE0050
 6    K=2                                                          NUCE0060
      CALL INITAS (X)                                              NUCE0070
      GØ TØ 10                                                     NUCE0080
 8    TYPE 1                                                       NUCE0090
 1    FØRMAT(18HNØ MØRE FREE SPACE)                                NUCE0100
      CALL EXIT                                                    NUCE0110
 2    IF(ID(CØNT(M))-1)12,14,12                                    NUCE0120
 14   CALL IRALST (CØNT(M+1))                                      NUCE0130
 12   CALL SETIND (-1,-1,LNKR(CØNT(M)),-1)                        NUCE0140
      CALL STRIND (0,M)                                            NUCE0150
      CALL STRIND (0,M+1)                                          NUCE0160
      NUCELL=M                                                     NUCE0170
      RETURN                                                       NUCE0180
      END                                                          NUCE0190
```

```
ZZ    RCELL  0655

ZZFØR

*FANDK0810
*LDISKRCELL 0655
      SUBRØUTINE RCELL (CELL)                                    RCEL0010
      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                                                         RCEL0060
```

```
ZZ    MTLIST 0657

ZZFØR

*FANDK0810
*LDISKMTLIST0657
      FUNCTIØN MTLIST (LST)                                    MTLI0010
      L=LØCT(LST)                                              MTLI0020
      IF(LISTMT(L)) 2,4,2                                      MTLI0030
 2    LR=LNKR(CØNT(L))                                         MTLI0040
      LL=LNKL(CØNT(L))                                         MTLI0050
      CALL SETIND (-1,L,L,L)                                   MTLI0060
      CALL SETIND(-1,-1,LR,LNKL(CØNT(-1)))                     MTLI0070
      CALL SETIND(-1,LL,-1,-1)                                 MTLI0080
      CALL SETIND(-1,-1,0,LL)                                  MTLI0090
 4    MTLIST=L                                                 MTLI0100
      RETURN                                                   MTLI0110
      END                                                      MTLI0120
```

```
ZZ    IRALST 0659

ZZFØR

*FANDK0810
*LDISKIRALST0659
      FUNCTIØN IRALST (LST)                                    IRAL0010
      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
      IF(N)6,8,6                                               IRAL0080
 6    CALL SETIND(1,-1,-1,L)                                   IRAL0100
      CALL STRIND (N,L+1)                                      IRAL0110
 8    CALL RCELL(L)                                            IRAL0130
 2    RETURN                                                   IRAL0140
      END                                                      IRAL0150
```

```
ZZ    LDUMP  0661

ZZFØR

*FANDK0810
*LDISKLDUMP 0661
      FUNCTIØN   LDUMP (N1,N2)                                          LDUM0010
      EQUIVALENCE (XW,IW)                                               LDUM0011
      IT=2-N1+N1/2*2                                                    LDUM0012
C     IT.. 1=ØDD PØSITIVE, 2=EVEN, 3=ØDD NEGATIVE.                      LDUM0014
      LL=LNKL(CØNT(-1))                                                 LDUM0016
      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
     1       10H     AVSL=,2I6)                                        LDUM0031
      LDUMP=0                                                           LDUM0040
      GØ TØ (4,6,6),IT                                                  LDUM0042
 6    PUNCH 5                                                           LDUM0044
 5    FØRMAT(65H0   ADDRESS ØF FIRST CELL MUST BE ØDD AND PØSITIVE--LDUMLDUM0046
     1P ØMITTED)                                                       LDUM0047
      RETURN                                                            LDUM0050
 4    DØ 2 I=N1,N2,2                                                    LDUM0060
      LL=LNKL(CØNT(I))                                                  LDUM0080
      LR=LNKR(CØNT(I))                                                  LDUM0090
      IX=ID(CØNT(I))                                                    LDUM0100
      XW=CØNT(I+1)                                                      LDUM0120
      LDUMP=LDUMP+1                                                     LDUM0122
 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 6HDATUM=,I11,2X,2HØR,E16.10)                                    LDUM0150
      RETURN                                                            LDUM0160
      END                                                               LDUM0200
```

```
ZZ    IQUAL  0665

ZZFØR

*FANDK0810
*LDISKIQUAL 0665
      FUNCTIØN IQUAL (K,L)                                    IQUA0010
      IF(LNKL(K)-LNKL(L))2,4,2                                IQUA0020
 4    IF(LNKR(K)-LNKR(L))2,6,2                                IQUA0030
 6    IF(ID(K)-ID(L))2,8,2                                    IQUA0040
C8    IQUAL=K-L                                               IQUA0050
 8    IQUAL=0                                                 IQUA0050
      RETURN                                                  IQUA0060
 2    IQUAL=1                                                 IQUA0070
      RETURN                                                  IQUA0080
      END                                                     IQUA0090
```

```
ZZ    LISTMT 0667

ZZFØR

*FANDK0810
*LDISKLISTMT0667
      FUNCTIØN LISTMT (LST)                                    LIST0010
      L=LØCT(LST)                                              LIST0020
      LISTMT=IQUAL(CØNT(L),CØNT(LNKR(CØNT(L))))                LIST0030
      RETURN                                                   LIST0040
      END                                                      LIST0050
```

```
ZZ    NAMTST 0669

ZZFØR

*FANDK0810
*LDISKNAMTST0669
      FUNCTIØN NAMTST (L)                                                    NAMT0010
      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
      RETURN                                                                 NAMT0060
 2    NAMTST=1                                                               NAMT0070
      RETURN                                                                 NAMT0080
      END                                                                    NAMT0090
```

```
ZZ     LØCT   0671

ZZFØR

*FANDK0810
*LDISKLØCT  0671
      FUNCTIØN LØCT (LST)                                          LØCT0010
      IF(NAMTST(LST))2,4,2                                         LØCT0020
 4    LØCT=LST                                                     LØCT0030
      RETURN                                                       LØCT0040
 2    TYPE 1,LST                                                   LØCT0050
 1    FØRMAT(16HØPERAND NØT LIST,I11)                              LØCT0060
      CALL EXIT                                                    LØCT0070
      END                                                          LØCT0080

$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE -- D.P.K.
```

```
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
           DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
           DSC  17,0,0
           DØRGQ-100
     FAC   DS   ,2492
     FXZ   DS   ,3099
     BETA  DS   ,2630
     IPNT  DS   5
     ICHAR DSA
     WØRD  DSA
     LST   DSA
           DC   1,'
           DS   5
     ENT   TFM  TF+6,ICHAR-4
           AM   TF+6,4,10
           AM   ENT-1,5,10
           TF   CF+11,1-ENT
           BNF  *+36,CF+11
     CF    CF   CF+11
           TF   CF+11,-CF-11
     TF    TF   ,CF+11
           AM   TF+6,1,10
           BNR  ENT+12,-TF-6
           AM   ENT-1,2,10
           BSBA*+12
           CM   -WØRD,,10
           BNE  *+24
           TFM  IPNT
           AM   IPNT,2,10
           CM   IPNT,12,10
           BNE  NØT
```

```
            BTM  ,*+11
NEWBØTDS    ,*-5
            DSA  -WØRD
            DSA  -LST
            TF   -WØRD,FXZ
            TFM  IPNT,2,10
NØT         BLX  *+12,IPNT(A1)
            TF   FAC,-WØRD
            TF   BETA,-ICHAR
            TD   FAC-11(A1),BETA-1
            TD   FAC-10(A1),BETA
            SF   FAC-9
            TF   -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.
```

```
ZZ    LIST   0675

ZZFØR

*FANDK0810
*LDISKLIST  0675
      FUNCTIØN 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
```

```
ZZ    TØP    0677

ZZFØR

*FANDK0810
*LDISKTØP   0677
      FUNCTIØN TØP(P)
      TØP=CØNT(LNKR(CØNT(LØCT(P))))+1)
      RETURN
      END
```

```
ZZ     BØT     0679

ZZFØR

*FANDK0810
*LDISKBØT    0679
       FUNCTIØN BØT(P)
       BØT=CØNT(LNKL(CØNT(LØCT(P))))+1)
       RETURN
       END
```

```
ZZ    PØPTØP 0681

ZZFØR

*FANDK0810
*LDISKPØPTØP0681
      FUNCTIØN PØPTØP(P)
      PØPTØP=DELETE(LNKR(CØNT(LØCT(P))))
      RETURN
      END
```

```
ZZ    PØPBØT 0683

ZZFØR

*FANDK0810
*LDISKPØPBØT0683
      FUNCTIØN PØPBØT(P)
      PØPBØT=DELETE(LNKL(CØNT(LØCT(P))))
      RETURN
      END
```

```
ZZ     NEWTØP 0685

ZZFØR

*FANDK0810
*LDISKNEWTØP0685
       FUNCTIØN NEWTØP(P,Q)
       NEWTØP=NXTRGT(P,LØCT(Q))
       RETURN
       END
```

```
ZZ    NEWBØT 0687

ZZFØR

*FANDK0810
*LDISKNEWBØT0687
      FUNCTIØN NEWBØT(P,Q)
      NEWBØT=NXTLFT(P,LØCT(Q))
      RETURN
      END
```

```
ZZ    NXTRGT 0689

ZZFØR

*FANDK0810
*LDISKNXTRGT0689
      FUNCTIØN NXTRGT(M,A)
      IR=NUCELL(Z)
      NXTRGT=IR
      LR=LNKR(CØNT(A))
      CALL SETIND(-1,IR,-1,LR)
      CALL SETIND(-1,-1,IR,A)
      CALL SETIND(0,A,LR,IR)
      IF(NAMTST(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
      END
```

```
ZZ    NXTLFT 0691

ZZFØR

*FANDK0810
*LDISKNXTLFT0691
      FUNCTIØN NXTLFT(M,A)
      IL=NUCELL(Z)
      NXTLFT=IL
      LL=LNKL(CØNT(A))
      CALL SETIND(-1,-1,IL,LL)
      CALL SETIND(-1,IL,-1,A)
      CALL SETIND(0,LL,A,IL)
      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)
      RETURN
      END
```

```
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)
      LL=LNKL(CØNT(K))
      LR=LNKR(CØNT(K))
      CALL RCELL(K)
      CALL SETIND(-1,-1,LR,LL)
      CALL SETIND(-1,LL,-1,LR)
      RETURN
      END
```

```
ZZ    LISTRD 0695

ZZFØR

*FANDK0810
*LDISKLISTRD0695
      FUNCTIØN 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
      CALL NEWTØP(LST,LIST(STACK))
   40 IDELSW=0
      NUM=0
      IFLTSW=0
      ISINSW=0
   50 ICHAR=KGET(IN,ICØDE)
      IF (ICØDE) 51,51,54
C END ØF LIST
   51 IF(IDELSW)53,53,52
   52 CALL NEWBØT(WØRD,TØP(STACK))
   53 CALL IRALST(STACK)
      RETURN
   54 GØ TØ (110,120,130,140,150,160,170,180,190,200,210,220,230,
     1  240),ICØDE
C EQUAL SIGN
  110 IF(IDELSW)112,112,111
  111 CALL NEWBØT(WØRD,TØP(STACK))
  112 NUM=0
      CALL KPUT(ICHAR,WØRD,TØP(STACK))
      CALL NEWBØT(WØRD,TØP(STACK))
      GØ TØ 40
C MINUS SIGN
  120 IF(IDELSW)121,121,111
  121 IDELSW=1
      ISINSW=-1
      CALL KPUT(ICHAR,WØRD,TØP(STACK))
      GØ TØ 50
C PLUS SIGN
  130 IF (IDELSW)131,131,111
  131 IDELSW=1
```

```
            ISINSW=1
            CALL KPUT(ICHAR,WØRD,TØP(STACK))
            GØ TØ 50
      C LETTER
        140 IDELSW=1
            CALL KPUT(ICHAR,WØRD,TØP(STACK))
            GØ TØ 50
      C DECIMAL PØINT
        150 IF(IDELSW)112,112,152
        152 IF(IFLTSW)153,111,111
        153 WØRD=NUM
            SIGN=ISIGN
            IFLTSW=1
            DEC=10.
            GØ TØ 50
      C RIGHT PAREN
        160 IF(IDELSW)162,162,161
        161 CALL NEWBØT(WØRD,TØP(STACK))
        162 CALL PØPTØP(STACK)
            ISUB=ISUB-1
            IF(ISUB)163,163,40
        163 CALL IRALST(STACK)
            RETURN
      C DØLLAR SIGN
        170 GØ TØ 110
      C ASTERISK
        180 GØ TØ 110
      C BLANK
        190 IF(IDELSW)50,50,191
        191 CALL NEWBØT(WØRD,TØP(STACK))
            GØ TØ 40
      C LEFT PAREN
        200 IF(IDELSW)202,202,201
        201 CALL NEWBØT(WØRD,TØP(STACK))
        202 ISUB=ISUB+1
            CALL NEWBØT(LIST(9),TØP(STACK))
            CALL NEWTØP(BØT(TØP(STACK)),STACK)
            GØ TØ 40
      C SLASH
        210 GØ TØ 110
      C NUMBER
        220 IF(ISINSW)223,221,223
        221 IF(IDELSW)222,222,224
        222 ISINSW=1
```

```
  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
      WØRD=WØRD+SIGN*VAL/DEC
      DEC=DEC*10.
      GØ TØ 50
C CØMMA
  230 GØ TØ 110
C APPØSTRØPHE
  240 GØ TØ 140
      END
```

```
ZZFØR

ZZ     SEQRDR 0697

*FANDK0810
*LDISKSEQRDR0697
       FUNCTION SEQRDR(LST)
       SEQRDR=CØNT(LØCT(LST))
       RETURN
       END
```

```
ZZ    SEQLR  0699

ZZFØR

*FANDK0810
*LDISKSEQLR 0699
      FUNCTIØN SEQLR(Z,N)
      L=LNKR(Z)
      Z=CØNT(L)
      SEQLR=CØNT(L+1)
      N=ID(Z)-1
      RETURN
      END
```

```
ZZ     SEQLL  0701

ZZFØR

*FANDK0810
*LDISKSEQLL 0701
      FUNCTIØN SEQLL(Z,N)
      L=LNKL(Z)
      Z=CØNT(L)
      SEQLL=CØNT(L+1)
      N=ID(Z)-1
      RETURN
      END
```

```
ZZ    SEQSR  0703

ZZFØR

*FANDK0810
*LDISKSEQSR 0703
      FUNCTIØN SEQSR(Z,N)
      IF(ID(Z)-1)4,5,4
    5 L=LNKR(CØNT(CØNT(LNKL(CØNT(LNKR(Z)))+1)))
      GØ TØ 3
    4 L=LNKR(Z)
    3 IF(ID(CØNT(L))-1)1,2,1
    1 SEQSR=CØNT(L+1)
      Z=CØNT(L)
      N=ID(Z)-1
      RETURN
    2 L=LNKR(CØNT(CØNT(L+1)))
      GØ TØ 3
      END
```

```
ZZ     SEQSL  0705

ZZFØR

*FANDK0810
*LDISKSEQSL 0705
      FUNCTIØN SEQSL(Z,N)
      IF(ID(Z)-1)4,5,4
    5 L=LNKL(CØNT(CØNT(LNKL(CØNT(LNKR(Z)))+1)))
      GØ TØ 3
    4 L=LNKL(Z)
    3 IF(ID(CØNT(L))-1)1,2,1
    1 SEQSL=CØNT(L+1)
      Z=CØNT(L)
      N=ID(Z)-1
      RETURN
    2 L=LNKL(CØNT(CØNT(L+1)))
      GØ TØ 3
      END

$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE -- R.L.S.
```

SAMPLE PROGRAM + DATA

```
ZZJØB                                  292701265073 87+215

ZZFØRX
*FANDK0810
      B=0.
      L=LIST(B)
   10 CALL LISTRD(L,5)
      I=NUCELL(L)
      CALL LDUMP(1,I)
      CALL RCELL(I)
      GØ TØ 10
      END
(123)
(A B C D)
( A B ( C D ))
(123.456)
(A+Ø./ C.D,S B +5 -4 A'BCDEFGHI A$BC )
(
      FUNCTIØN   LDUMP (N1,N2)                                          LDUM0010
      EQUIVALENCE (XW,IW)                                               LDUM0011
      IT=2-N1+N1/2*2                                                    LDUM0012
C     IT.. 1=ØDD PØSITIVE, 2=EVEN, 3=ØDD NEGATIVE.                      LDUM0014
      LL=LNKL(CØNT(-1))                                                 LDUM0016
      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
     1     10H     AVSL=,2I6)                                          LDUM0031
      LDUMP=0                                                           LDUM0040
      GØ TØ (4,6,6),IT                                                  LDUM0042
 6    PUNCH 5                                                           LDUM0044
 5    FØRMAT(65H0   ADDRESS ØF FIRST CELL MUST BE ØDD AND PØSITIVE--LDUMLDUM0046
     1P ØMITTED)                                                       LDUM0047
      RETURN                                                            LDUM0050
 4    DØ 2 I=N1,N2,2                                                    LDUM0060
      LL=LNKL(CØNT(I))                                                  LDUM0080
      LR=LNKR(CØNT(I))                                                  LDUM0090
      IX=ID(CØNT(I))                                                    LDUM0100
      XW=CØNT(I+1)                                                      LDUM0120
      LDUMP=LDUMP+1                                                     LDUM0122
 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        6HDATUM=,I11,2X,2HØR,E16.10)                              LDUM0150
      RETURN                                                            LDUM0160
```

```
      END                                                    LDUM0200
     )
```

OUTPUT FROM SAMPLE PROGRAM

```
LDUMP OF SLIP-CELLS     1   THRU   669     AVSL=     3   671
  1   LNKL=  667   LNKR=    7   ID= 2   DATUM=           2  ØR   .20000000E+01
  3   LNKL=    3   LNKR=    0   ID= 2   DATUM=           0  ØR   .00000000E-99
  5   LNKL=    3   LNKR=    3   ID= 1   DATUM=      100001  ØR   .10000100E+06
  7   LNKL=    1   LNKR=    9   ID= 0   DATUM= 4664554363  ØR   .46645543E+10
  9   LNKL=    7   LNKR=   11   ID= 0   DATUM= 4956550000  ØR   .49565500E+10
 11   LNKL=    9   LNKR=   15   ID= 0   DATUM= 5344645457  ØR   .53446454E+10
 13   LNKL=   23   LNKR=   19   ID= 2   DATUM=           2  ØR   .20000000E+01
 15   LNKL=   11   LNKR=   25   ID= 1   DATUM=     1300013  ØR   .13000130E+07
 17   LNKL=    3   LNKR=   39   ID= 1   DATUM=     1300013  ØR   .13000130E+07
 19   LNKL=   13   LNKR=   21   ID= 0   DATUM= 5571000000  ØR   .55710000E+10
 21   LNKL=   19   LNKR=   23   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
 23   LNKL=   21   LNKR=   13   ID= 0   DATUM= 5572000000  ØR   .55720000E+10
 25   LNKL=   15   LNKR=   27   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
 27   LNKL=   25   LNKR=   29   ID= 0   DATUM= 7071700000  ØR   .70717000E+10
 29   LNKL=   27   LNKR=   31   ID= 0   DATUM= 4558644965  ØR   .45586449E+10
 31   LNKL=   29   LNKR=   33   ID= 0   DATUM= 4153455543  ØR   .41534555E+10
 33   LNKL=   31   LNKR=   37   ID= 0   DATUM= 4500000000  ØR   .45000000E+10
 35   LNKL=   45   LNKR=   41   ID= 2   DATUM=           2  ØR   .20000000E+01
 37   LNKL=   33   LNKR=   47   ID= 1   DATUM=     3500035  ØR   .35000350E+07
 39   LNKL=    3   LNKR=  139   ID= 1   DATUM=     3500035  ØR   .35000350E+07
 41   LNKL=   35   LNKR=   43   ID= 0   DATUM= 6766000000  ØR   .67660000E+10
 43   LNKL=   41   LNKR=   45   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
 45   LNKL=   43   LNKR=   35   ID= 0   DATUM= 4966000000  ØR   .49660000E+10
 47   LNKL=   37   LNKR=   49   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
 49   LNKL=   47   LNKR=   51   ID= 0   DATUM= 7071710000  ØR   .70717100E+10
 51   LNKL=   49   LNKR=   53   ID= 0   DATUM= 4963000000  ØR   .49630000E+10
 53   LNKL=   51   LNKR=   55   ID= 0   DATUM= 3300000000  ØR   .33000000E+10
 55   LNKL=   53   LNKR=   57   ID= 0   DATUM=           2  ØR   .20000000E+01
 57   LNKL=   55   LNKR=   59   ID= 0   DATUM= 2000000000  ØR   .20000000E+10
 59   LNKL=   57   LNKR=   61   ID= 0   DATUM= 5571000000  ØR   .55710000E+10
 61   LNKL=   59   LNKR=   63   ID= 0   DATUM= 1000000000  ØR   .10000000E+10
 63   LNKL=   61   LNKR=   65   ID= 0   DATUM= 5571000000  ØR   .55710000E+10
 65   LNKL=   63   LNKR=   67   ID= 0   DATUM= 2100000000  ØR   .21000000E+10
 67   LNKL=   65   LNKR=   69   ID= 0   DATUM=           2  ØR   .20000000E+01
 69   LNKL=   67   LNKR=   71   ID= 0   DATUM= 1400000000  ØR   .14000000E+10
 71   LNKL=   69   LNKR=   73   ID= 0   DATUM=           2  ØR   .20000000E+01
 73   LNKL=   71   LNKR=   75   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
 75   LNKL=   73   LNKR=   77   ID= 0   DATUM= 7071720000  ØR   .70717200E+10
 77   LNKL=   75   LNKR=   79   ID= 0   DATUM= 4300000000  ØR   .43000000E+10
 79   LNKL=   77   LNKR=   81   ID= 0   DATUM= 4963000000  ØR   .49630000E+10
 81   LNKL=   79   LNKR=   83   ID= 0   DATUM=  300000000  ØR   .30000000E+09
 83   LNKL=   81   LNKR=   85   ID= 0   DATUM=  300000000  ØR   .30000000E+09
```

85	LNKL=	83	LNKR=	87	ID= 0	DATUM=	1	ØR	.10000000E+01
87	LNKL=	85	LNKR=	89	ID= 0	DATUM=	3300000000	ØR	.33000000E+10
89	LNKL=	87	LNKR=	91	ID= 0	DATUM=	5644440000	ØR	.56444400E+10
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=	2300000000	ØR	.23000000E+10
97	LNKL=	95	LNKR=	99	ID= 0	DATUM=	2	ØR	.20000000E+01
99	LNKL=	97	LNKR=	101	ID= 0	DATUM=	3300000000	ØR	.33000000E+10
101	LNKL=	99	LNKR=	103	ID= 0	DATUM=	4565455500	ØR	.45654555E+10
103	LNKL=	101	LNKR=	105	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
105	LNKL=	103	LNKR=	107	ID= 0	DATUM=	3	ØR	.30000000E+01
107	LNKL=	105	LNKR=	109	ID= 0	DATUM=	3300000000	ØR	.33000000E+10
109	LNKL=	107	LNKR=	111	ID= 0	DATUM=	5644440000	ØR	.56444400E+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=	300000000	ØR	.30000000E+09
117	LNKL=	115	LNKR=	119	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
119	LNKL=	117	LNKR=	121	ID= 0	DATUM=	7071740000	ØR	.70717400E+10
121	LNKL=	119	LNKR=	123	ID= 0	DATUM=	5353000000	ØR	.53530000E+10
123	LNKL=	121	LNKR=	125	ID= 0	DATUM=	3300000000	ØR	.33000000E+10
125	LNKL=	123	LNKR=	129	ID= 0	DATUM=	5355525300	ØR	.53555253E+10
127	LNKL=	137	LNKR=	133	ID= 2	DATUM=	2	ØR	.20000000E+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
135	LNKL=	141	LNKR=	141	ID= 2	DATUM=	2	ØR	.20000000E+01
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	-.10000000E+01
143	LNKL=	129	LNKR=	145	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
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=	3300000000	ØR	.33000000E+10
151	LNKL=	149	LNKR=	155	ID= 0	DATUM=	5355525900	ØR	.53555259E+10
153	LNKL=	163	LNKR=	159	ID= 2	DATUM=	2	ØR	.20000000E+01
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
161	LNKL=	167	LNKR=	167	ID= 2	DATUM=	2	ØR	.20000000E+01
163	LNKL=	159	LNKR=	153	ID= 1	DATUM=	16100161	ØR	.16100161E+08
165	LNKL=	3	LNKR=	157	ID= 1	DATUM=	16100161	ØR	.16100161E+08
167	LNKL=	161	LNKR=	161	ID= 0	DATUM=	-1	ØR	-.10000000E+01
169	LNKL=	155	LNKR=	171	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
171	LNKL=	169	LNKR=	173	ID= 0	DATUM=	7071780000	ØR	.70717800E+10

173	LNKL=	171	LNKR=	175	ID= 0	DATUM=	5764554348	ØR	.57645543E+10
175	LNKL=	173	LNKR=	177	ID= 0	DATUM=	3	ØR	.30000000E+01
177	LNKL=	175	LNKR=	179	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
179	LNKL=	177	LNKR=	181	ID= 0	DATUM=	4963000000	ØR	.49630000E+10
181	LNKL=	179	LNKR=	183	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
183	LNKL=	181	LNKR=	185	ID= 0	DATUM=	5571000000	ØR	.55710000E+10
185	LNKL=	183	LNKR=	187	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
187	LNKL=	185	LNKR=	189	ID= 0	DATUM=	5572000000	ØR	.55720000E+10
189	LNKL=	187	LNKR=	191	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
191	LNKL=	189	LNKR=	193	ID= 0	DATUM=	5353000000	ØR	.53530000E+10
193	LNKL=	191	LNKR=	195	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
195	LNKL=	193	LNKR=	197	ID= 0	DATUM=	5359000000	ØR	.53590000E+10
197	LNKL=	195	LNKR=	199	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
199	LNKL=	197	LNKR=	201	ID= 0	DATUM=	7072700000	ØR	.70727000E+10
201	LNKL=	199	LNKR=	203	ID= 0	DATUM=	3	ØR	.30000000E+01
203	LNKL=	201	LNKR=	205	ID= 0	DATUM=	4656595441	ØR	.46565954E+10
205	LNKL=	203	LNKR=	209	ID= 0	DATUM=	6300000000	ØR	.63000000E+10
207	LNKL=	263	LNKR=	213	ID= 2	DATUM=	2	ØR	.20000000E+01
209	LNKL=	205	LNKR=	265	ID= 1	DATUM=	20700207	ØR	.20700207E+08
211	LNKL=	3	LNKR=	287	ID= 1	DATUM=	20700207	ØR	.20700207E+08
213	LNKL=	207	LNKR=	215	ID= 0	DATUM=	4971000000	ØR	.49710000E+10
215	LNKL=	213	LNKR=	217	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
217	LNKL=	215	LNKR=	219	ID= 0	DATUM=	67000005	ØR	.67000005E+08
219	LNKL=	217	LNKR=	221	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
221	LNKL=	219	LNKR=	223	ID= 0	DATUM=	48534464	ØR	.48534464E+08
223	LNKL=	221	LNKR=	225	ID= 0	DATUM=	5457000000	ØR	.54570000E+10
225	LNKL=	223	LNKR=	227	ID= 0	DATUM=	5646000000	ØR	.56460000E+10
227	LNKL=	225	LNKR=	229	ID= 0	DATUM=	6253495700	ØR	.62534957E+10
229	LNKL=	227	LNKR=	231	ID= 0	DATUM=	2000000000	ØR	.20000000E+10
231	LNKL=	229	LNKR=	233	ID= 0	DATUM=	4345535362	ØR	.43455353E+10
233	LNKL=	231	LNKR=	235	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
235	LNKL=	233	LNKR=	237	ID= 0	DATUM=	4976000000	ØR	.49760000E+10
237	LNKL=	235	LNKR=	239	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
239	LNKL=	237	LNKR=	241	ID= 0	DATUM=	48000007	ØR	.48000007E+08
241	LNKL=	239	LNKR=	243	ID= 0	DATUM=	6348596400	ØR	.63485964E+10
243	LNKL=	241	LNKR=	245	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
245	LNKL=	243	LNKR=	247	ID= 0	DATUM=	4976000000	ØR	.49760000E+10
247	LNKL=	245	LNKR=	249	ID= 0	DATUM=	2300000000	ØR	.23000000E+10
249	LNKL=	247	LNKR=	251	ID= 0	DATUM=	5344645470	ØR	.53446454E+10
251	LNKL=	249	LNKR=	253	ID= 0	DATUM=	7073700000	ØR	.70737000E+10
253	LNKL=	251	LNKR=	255	ID= 0	DATUM=	1	ØR	.10000000E+01
255	LNKL=	253	LNKR=	257	ID= 0	DATUM=	4810	ØR	.48100000E+04
257	LNKL=	255	LNKR=	259	ID= 0	DATUM=	4165625300	ØR	.41656253E+10

259	LNKL=	257	LNKR=	261	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
261	LNKL=	259	LNKR=	263	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
263	LNKL=	261	LNKR=	207	ID=	0	DATUM=	490000026	ØR	.49000002E+09
265	LNKL=	209	LNKR=	267	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
267	LNKL=	265	LNKR=	269	ID=	0	DATUM=	7073710000	ØR	.70737100E+10
269	LNKL=	267	LNKR=	271	ID=	0	DATUM=	5344645457	ØR	.53446454E+10
271	LNKL=	269	LNKR=	273	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
273	LNKL=	271	LNKR=	275	ID=	0	DATUM=	0	ØR	.00000000E-99
275	LNKL=	273	LNKR=	277	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
277	LNKL=	275	LNKR=	279	ID=	0	DATUM=	7074700000	ØR	.70747000E+10
279	LNKL=	277	LNKR=	281	ID=	0	DATUM=	4756000000	ØR	.47560000E+10
281	LNKL=	279	LNKR=	285	ID=	0	DATUM=	6356000000	ØR	.63560000E+10
283	LNKL=	297	LNKR=	289	ID=	2	DATUM=	2	ØR	.20000000E+01
285	LNKL=	281	LNKR=	299	ID=	1	DATUM=	28300283	ØR	.28300283E+08
287	LNKL=	3	LNKR=	327	ID=	1	DATUM=	28300283	ØR	.28300283E+08
289	LNKL=	283	LNKR=	291	ID=	0	DATUM=	4	ØR	.40000000E+01
291	LNKL=	289	LNKR=	293	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
293	LNKL=	291	LNKR=	295	ID=	0	DATUM=	6	ØR	.60000000E+01
295	LNKL=	293	LNKR=	297	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
297	LNKL=	295	LNKR=	283	ID=	0	DATUM=	6	ØR	.60000000E+01
299	LNKL=	285	LNKR=	301	ID=	0	DATUM=	2300000000	ØR	.23000000E+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	.60000000E+01
309	LNKL=	307	LNKR=	311	ID=	0	DATUM=	5764554348	ØR	.57645543E+10
311	LNKL=	309	LNKR=	313	ID=	0	DATUM=	5	ØR	.50000000E+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	.50000000E+01
319	LNKL=	317	LNKR=	321	ID=	0	DATUM=	4656595441	ØR	.46565954E+10
321	LNKL=	319	LNKR=	325	ID=	0	DATUM=	6300000000	ØR	.63000000E+10
323	LNKL=	363	LNKR=	329	ID=	2	DATUM=	2	ØR	.20000000E+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
329	LNKL=	323	LNKR=	331	ID=	0	DATUM=	480000650	ØR	.48000065E+09
331	LNKL=	329	LNKR=	333	ID=	0	DATUM=	4144445945	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

347	LNKL=	345	LNKR=	349	ID=	0	DATUM=	4155440000	ØR	.41554400E+10
349	LNKL=	347	LNKR=	351	ID=	0	DATUM=	5756624963	ØR	.57566249E+10
351	LNKL=	349	LNKR=	353	ID=	0	DATUM=	4965450000	ØR	.49654500E+10
353	LNKL=	351	LNKR=	355	ID=	0	DATUM=	2000000000	ØR	.20000000E+10
355	LNKL=	353	LNKR=	357	ID=	0	DATUM=	2053446454	ØR	.20534464E+10
357	LNKL=	355	LNKR=	359	ID=	0	DATUM=	-46	ØR	-.46000000E+02
359	LNKL=	357	LNKR=	361	ID=	0	DATUM=	57	ØR	.57000000E+02
361	LNKL=	359	LNKR=	363	ID=	0	DATUM=	5654496363	ØR	.56544963E+10
363	LNKL=	361	LNKR=	323	ID=	0	DATUM=	4544000000	ØR	.45440000E+10
365	LNKL=	325	LNKR=	367	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
367	LNKL=	365	LNKR=	369	ID=	0	DATUM=	7074770000	ØR	.70747700E+10
369	LNKL=	367	LNKR=	371	ID=	0	DATUM=	5945636459	ØR	.59456364E+10
371	LNKL=	369	LNKR=	373	ID=	0	DATUM=	5500000000	ØR	.55000000E+10
373	LNKL=	371	LNKR=	375	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
375	LNKL=	373	LNKR=	377	ID=	0	DATUM=	7075700000	ØR	.70757000E+10
377	LNKL=	375	LNKR=	379	ID=	0	DATUM=	4	ØR	.40000000E+01
379	LNKL=	377	LNKR=	381	ID=	0	DATUM=	4456000000	ØR	.44560000E+10
381	LNKL=	379	LNKR=	383	ID=	0	DATUM=	2	ØR	.20000000E+01
383	LNKL=	381	LNKR=	385	ID=	0	DATUM=	4900000000	ØR	.49000000E+10
385	LNKL=	383	LNKR=	387	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
387	LNKL=	385	LNKR=	389	ID=	0	DATUM=	5571000000	ØR	.55710000E+10
389	LNKL=	387	LNKR=	391	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
391	LNKL=	389	LNKR=	393	ID=	0	DATUM=	5572000000	ØR	.55720000E+10
393	LNKL=	391	LNKR=	395	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
395	LNKL=	393	LNKR=	397	ID=	0	DATUM=	2	ØR	.20000000E+01
397	LNKL=	395	LNKR=	399	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
399	LNKL=	397	LNKR=	401	ID=	0	DATUM=	7076700000	ØR	.70767000E+10
401	LNKL=	399	LNKR=	403	ID=	0	DATUM=	5353000000	ØR	.53530000E+10
403	LNKL=	401	LNKR=	405	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
405	LNKL=	403	LNKR=	409	ID=	0	DATUM=	5355525300	ØR	.53555253E+10
407	LNKL=	417	LNKR=	413	ID=	2	DATUM=	2	ØR	.20000000E+01
409	LNKL=	405	LNKR=	423	ID=	1	DATUM=	40700407	ØR	.40700407E+08
411	LNKL=	3	LNKR=	445	ID=	1	DATUM=	40700407	ØR	.40700407E+08
413	LNKL=	407	LNKR=	417	ID=	0	DATUM=	4356556300	ØR	.43565563E+10
415	LNKL=	421	LNKR=	421	ID=	2	DATUM=	2	ØR	.20000000E+01
417	LNKL=	413	LNKR=	407	ID=	1	DATUM=	41500415	ØR	.41500415E+08
419	LNKL=	3	LNKR=	411	ID=	1	DATUM=	41500415	ØR	.41500415E+08
421	LNKL=	415	LNKR=	415	ID=	0	DATUM=	4900000000	ØR	.49000000E+10
423	LNKL=	409	LNKR=	425	ID=	0	DATUM=	5344645470	ØR	.53446454E+10
425	LNKL=	423	LNKR=	427	ID=	0	DATUM=	7078700000	ØR	.70787000E+10
427	LNKL=	425	LNKR=	429	ID=	0	DATUM=	5359000000	ØR	.53590000E+10
429	LNKL=	427	LNKR=	431	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
431	LNKL=	429	LNKR=	435	ID=	0	DATUM=	5355525900	ØR	.53555259E+10

```
433   LNKL=  443   LNKR=  439   ID= 2   DATUM=          2  ØR   .20000000E+01
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
441   LNKL=  447   LNKR=  447   ID= 2   DATUM=          2  ØR   .20000000E+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   .49000000E+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= 4967000000  ØR   .49670000E+10
455   LNKL=  453   LNKR=  457   ID= 0   DATUM= 3300000000  ØR   .33000000E+10
457   LNKL=  455   LNKR=  461   ID= 0   DATUM= 4944000000  ØR   .49440000E+10
459   LNKL=  469   LNKR=  465   ID= 2   DATUM=          2  ØR   .20000000E+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
467   LNKL=  473   LNKR=  473   ID= 2   DATUM=          2  ØR   .20000000E+01
469   LNKL=  465   LNKR=  459   ID= 1   DATUM=   46700467  ØR   .46700467E+08
471   LNKL=    3   LNKR=  463   ID= 1   DATUM=   46700467  ØR   .46700467E+08
473   LNKL=  467   LNKR=  467   ID= 0   DATUM= 4900000000  ØR   .49000000E+10
475   LNKL=  461   LNKR=  477   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
477   LNKL=  475   LNKR=  479   ID= 0   DATUM= 7170700000  ØR   .71707000E+10
479   LNKL=  477   LNKR=  481   ID= 0   DATUM= 6766000000  ØR   .67660000E+10
481   LNKL=  479   LNKR=  483   ID= 0   DATUM= 3300000000  ØR   .33000000E+10
483   LNKL=  481   LNKR=  487   ID= 0   DATUM= 4356556300  ØR   .43565563E+10
485   LNKL=  495   LNKR=  491   ID= 2   DATUM=          2  ØR   .20000000E+01
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= 4900000000  ØR   .49000000E+10
493   LNKL=  491   LNKR=  495   ID= 0   DATUM= 1000000000  ØR   .10000000E+10
495   LNKL=  493   LNKR=  485   ID= 0   DATUM=          1  ØR   .10000000E+01
497   LNKL=  487   LNKR=  499   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
499   LNKL=  497   LNKR=  501   ID= 0   DATUM= 7172700000  ØR   .71727000E+10
501   LNKL=  499   LNKR=  503   ID= 0   DATUM= 5344645457  ØR   .53446454E+10
503   LNKL=  501   LNKR=  505   ID= 0   DATUM= 3300000000  ØR   .33000000E+10
505   LNKL=  503   LNKR=  507   ID= 0   DATUM= 5344645457  ØR   .53446454E+10
507   LNKL=  505   LNKR=  509   ID= 0   DATUM= 1000000000  ØR   .10000000E+10
509   LNKL=  507   LNKR=  511   ID= 0   DATUM=          1  ØR   .10000000E+01
511   LNKL=  509   LNKR=  513   ID= 0   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   .20000000E+01
517   LNKL=  515   LNKR=  519   ID= 0   DATUM= 5764554348  ØR   .57645543E+10
519   LNKL=  517   LNKR=  521   ID= 0   DATUM=          1  ØR   .10000000E+01
```

521	LNKL=	519	LNKR=	523	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
523	LNKL=	521	LNKR=	525	ID=	0	DATUM=	4900000000	ØR	.49000000E+10
525	LNKL=	523	LNKR=	527	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
527	LNKL=	525	LNKR=	529	ID=	0	DATUM=	5353000000	ØR	.53530000E+10
529	LNKL=	527	LNKR=	531	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
531	LNKL=	529	LNKR=	533	ID=	0	DATUM=	5359000000	ØR	.53590000E+10
533	LNKL=	531	LNKR=	535	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
535	LNKL=	533	LNKR=	537	ID=	0	DATUM=	4967000000	ØR	.49670000E+10
537	LNKL=	535	LNKR=	539	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
539	LNKL=	537	LNKR=	541	ID=	0	DATUM=	4966000000	ØR	.49660000E+10
541	LNKL=	539	LNKR=	543	ID=	0	DATUM=	2300000000	ØR	.23000000E+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
547	LNKL=	545	LNKR=	549	ID=	0	DATUM=	7173700000	ØR	.71737000E+10
549	LNKL=	547	LNKR=	551	ID=	0	DATUM=	1	ØR	.10000000E+01
551	LNKL=	549	LNKR=	553	ID=	0	DATUM=	4656595441	ØR	.46565954E+10
553	LNKL=	551	LNKR=	557	ID=	0	DATUM=	6300000000	ØR	.63000000E+10
555	LNKL=	649	LNKR=	561	ID=	2	DATUM=	2	ØR	.20000000E+01
557	LNKL=	553	LNKR=	651	ID=	1	DATUM=	55500555	ØR	.55500555E+08
559	LNKL=	3	LNKR=	5	ID=	1	DATUM=	55500555	ØR	.55500555E+08
561	LNKL=	555	LNKR=	563	ID=	0	DATUM=	67000001	ØR	.67000001E+08
563	LNKL=	561	LNKR=	565	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
565	LNKL=	563	LNKR=	567	ID=	0	DATUM=	4975000000	ØR	.49750000E+10
567	LNKL=	565	LNKR=	569	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
569	LNKL=	567	LNKR=	571	ID=	0	DATUM=	67000003	ØR	.67000003E+08
571	LNKL=	569	LNKR=	573	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
573	LNKL=	571	LNKR=	575	ID=	0	DATUM=	48535552	ØR	.48535552E+08
575	LNKL=	573	LNKR=	577	ID=	0	DATUM=	5300000000	ØR	.53000000E+10
577	LNKL=	575	LNKR=	579	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
579	LNKL=	577	LNKR=	581	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
581	LNKL=	579	LNKR=	583	ID=	0	DATUM=	4975000000	ØR	.49750000E+10
583	LNKL=	581	LNKR=	585	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
585	LNKL=	583	LNKR=	587	ID=	0	DATUM=	67000003	ØR	.67000003E+08
587	LNKL=	585	LNKR=	589	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
589	LNKL=	587	LNKR=	591	ID=	0	DATUM=	48535552	ØR	.48535552E+08
591	LNKL=	589	LNKR=	593	ID=	0	DATUM=	5900000000	ØR	.59000000E+10
593	LNKL=	591	LNKR=	595	ID=	0	DATUM=	3300000000	ØR	.33000000E+10
595	LNKL=	593	LNKR=	597	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
597	LNKL=	595	LNKR=	599	ID=	0	DATUM=	4975000000	ØR	.49750000E+10
599	LNKL=	597	LNKR=	601	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
601	LNKL=	599	LNKR=	603	ID=	0	DATUM=	67000003	ØR	.67000003E+08
603	LNKL=	601	LNKR=	605	ID=	0	DATUM=	2300000000	ØR	.23000000E+10
605	LNKL=	603	LNKR=	607	ID=	0	DATUM=	48494403	ØR	.48494403E+08

```
607   LNKL=  605   LNKR=  609   ID= 0   DATUM= 3300000000  ØR   .33000000E+10
609   LNKL=  607   LNKR=  611   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
611   LNKL=  609   LNKR=  613   ID= 0   DATUM= 4972000000  ØR   .49720000E+10
613   LNKL=  611   LNKR=  615   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
615   LNKL=  613   LNKR=  617   ID= 0   DATUM=   67000003  ØR   .67000003E+08
617   LNKL=  615   LNKR=  619   ID= 0   DATUM= 2300000000  ØR   .23000000E+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
623   LNKL=  621   LNKR=  625   ID= 0   DATUM=          1  ØR   .10000000E+01
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   .33000000E+10
631   LNKL=  629   LNKR=  633   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
633   LNKL=  631   LNKR=  635   ID= 0   DATUM= 4971710000  ØR   .49717100E+10
635   LNKL=  633   LNKR=  637   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
637   LNKL=  635   LNKR=  639   ID= 0   DATUM=   67000002  ØR   .67000002E+08
639   LNKL=  637   LNKR=  641   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
641   LNKL=  639   LNKR=  643   ID= 0   DATUM=   48565902  ØR   .48565902E+08
643   LNKL=  641   LNKR=  645   ID= 0   DATUM= 2300000000  ØR   .23000000E+10
645   LNKL=  643   LNKR=  647   ID= 0   DATUM= 4571760000  ØR   .45717600E+10
647   LNKL=  645   LNKR=  649   ID= 0   DATUM=  300000000  ØR   .30000000E+09
649   LNKL=  647   LNKR=  555   ID= 0   DATUM=         10  ØR   .10000000E+02
651   LNKL=  557   LNKR=  653   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
653   LNKL=  651   LNKR=  655   ID= 0   DATUM= 7175700000  ØR   .71757000E+10
655   LNKL=  653   LNKR=  657   ID= 0   DATUM= 5945636459  ØR   .59456364E+10
657   LNKL=  655   LNKR=  659   ID= 0   DATUM= 5500000000  ØR   .55000000E+10
659   LNKL=  657   LNKR=  661   ID= 0   DATUM= 5344645470  ØR   .53446454E+10
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
667   LNKL=  665   LNKR=    1   ID= 0   DATUM= 7270700000  ØR   .72707000E+10
669   LNKL=    0   LNKR=    0   ID= 0   DATUM=          0  ØR   .00000000E-99
```

```
‡‡JOB      0000400002          2927012650738 +215

‡‡FORX

*FANDK0810
0324 LENGTH
9999 NEXT COMMON
END OF COMPILATION
EXECUTION

MAIN  1200  0324  LOADED
LIST    1524  0574  LOADED
NUCELL  2098  1096  LOADED
LDUMP   3194  1750  LOADED
RCELL   4944  0612  LOADED
INITAS  5556  1028  LOADED
LISTRD  6584  3164  LOADED
SETDIR  9748  0374  LOADED
IQUAL   0122  0618  LOADED
SETIND  0740  0272  LOADED
LNKR    1012  0110  LOADED
CONT    1122  0158  LOADED
IRALST  1280  0822  LOADED
ID      2102  0158  LOADED
STRIND  2260  0242  LOADED
LNKL    2502  0122  LOADED
KGET    2624  1140  LOADED
NEWTOP  3764  0296  LOADED
NEWBOT  4060  0296  LOADED
TOP     4356  0362  LOADED
KPUT    4718  0384  LOADED
POPTOP  5102  0344  LOADED
BOT     5446  0362  LOADED
LOCT    5808  0352  LOADED
LCNTR   6160  0302  LOADED
MTLIST  6462  0948  LOADED
LSTNAM  7410  0350  LOADED
NXTRGT  7760  1014  LOADED
NXTLFT  8774  1014  LOADED
DELETE  9788  0900  LOADED
NAMTST  0688  0762  LOADED
LISTMT  1450  0436  LOADED
 PAGE 0 - 01399 SLIP CELLS INITIALIZED.
```

4-25-66

Kelleher and Sites

```
ZZ    LNKL   0630

ZZSPS
*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME LNKL
*ID NUMBER 0630
A1000****  LNKL   0630
A1010Q     DS   ,*+101
A1020      DC   6,987898,5-Q
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
A1060      DØRGQ-100
A1070FAC   DS   ,2492
A1080BETA  DS   ,2630
A1090FXZ   DS   ,3099
A1100      DS   5
A1110ENT   AM   *-1,5,10
A1120      TF   CF+11,1-ENT
A1130      BNF  TF,CF+11
A1140CF    CF   CF+11
A1150      TF   CF+11,-CF-11
A1160TF    AM   ENT-1,2,10
A1170      TFL  BETA,-CF-11
A1180      TF   FAC,FXZ
A1190      MA   FAC,BETA-5
A1200      B7   1-ENT
A1210LENGTHDC   2,'
A1220      DEND
```

```
ZZ    LNKR   0632

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME LNKR
*ID NUMBER 0632
B1000****  LNKR   0632
B1010Q     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
B1090      DS   5
B1100ENT   AM   *-1,5,10
B1110      TF   CF+11,1-ENT
B1120      BNF  TF,CF+11
B1130CF    CF   CF+11
B1140      TF   CF+11,-CF-11
B1150TF    AM   ENT-1,2,10
B1160      TF   FAC,FXZ
B1170      MA   FAC,-CF-11
B1180      B7   1-ENT
B1190LENGTHDC   2,'
B1200      DEND
```

```
ZZ    ID     0634

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME ID
*ID NUMBER 0634
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
C1070FAC   DS   ,2492
C1080BETA  DS   ,2630
C1090FXZ   DS   ,3099
C1100      DS   5
C1110ENT   AM   *-1,5,10
C1120      TF   CF+11,1-ENT
C1130      BNF  TF,CF+11
C1140CF    CF   CF+11
C1150      TF   CF+11,-CF-11
C1160TF    AM   ENT-1,2,10
C1170      TFL  BETA,-CF-11
C1180      TF   FAC,FXZ
C1190      BNF  *+24,BETA-1
C1192      AM   FAC,1,10
C1194      BNF  *+24,BETA-2
C1196      AM   FAC,2,10
C1200      B7   1-ENT
C1210LENGTHDC   2,'
C1220      DEND
```

```
ZZ    CØNT   0636

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME CØNT
*ID NUMBER 0636
D1000****  CØNT   0636
D1010Q     DS   ,*+101
D1020      DC   6,987898,5-Q
D1030      DAC  6,CØNT  ,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
D1080FXZ   DS   ,3099
D1082CALL  BTM  ,*+11
D1084INITASDS   ,*-5
D1090      DS   5
D1100ENT   B    CALL,,,TRANSFER VECTØR
D1105SLADR DS   ,*-5
D1109      AM   ENT-1,5,10
D1110      TF   CF+11,1-ENT
D1120      BNF  TF,CF+11
D1130CF    CF   CF+11
D1140      TF   CF+11,-CF-11
D1150TF    AM   ENT-1,2,10
D1160      MA   *+35,-CF-11
D1165      MF   *+23,-CF-11
D1170      BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
D1175      TFL  FAC,-99
D1180      B7   1-ENT
D1190QQ    DAC  6,INITAS,
D1192      DVLC,5,INITAS
D1194      DC   2,'
D1196LENGTHDS   ,QQ-1
D1200      DEND
```

```
ZZ    INITAS 0640

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME INITAS
*ID NUMBER 0640
E1000****  INITAS 0640
E1010Q     DS  ,*+101
E1020      DC  6,987898,5-Q
E1030      DAC 6,INITAS,7-Q
E1040      DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
E1050      DSC 17,0,0
E1060      DØRGQ-100
E1062FAC   DS  ,2492
E1063BETA  DS  ,2630
E1064FXZ   DS  ,3099
E1066FX1   DS  ,3109
E1070      DS  5
E1080ENT   AM  *-1,7,10
E1082      B7  *+8
E1084      TFM *-2,NEXT
E1090****  LINK TRANSFER VECTØRS TØ EXTERNALLY USED UTILITY ENTRY PØINTS
E1100      TRNM,SLADR.
E1110CØNT  DS  ,*-5
E1111      TRNM,SLADR.
E1112SETINDDS  ,*-5
E1113      TRNM,SLADR.
E1114STRINDDS  ,*-5
E1220****  CHAIN UP FREE SPACE - INITIALIZATIØN ØF CHAINER
E1230      TFM LWRLM,37986
E1235      TFM UPRLM,40002
E1240      TDM LWRLM,9
E1260      S   UPRLM,LWRLM
E1265      TF  DISK+8,UPRLM-2
E1267      TDM UPRLM-1
E1270      TDM UPRLM
E1280      A   UPRLM,LWRLM
E1283      AM  LWRLM,1,10
E1286      TF  DISK+13,LWRLM
E1289      AM  LWRLM,19,10
```

```
E1292       TF  AVSLL,AVSLR
E1294       SM  AVSLL,5,10
E1300****   PERFØRM CHAINING - MAY BE RE-USED BY ENTERING AT XXX
E1301****       AFTER SETTING BLKN AND RETN+6.
E1310XXX    BTAMSLADR,,78
E1312BLKN   DS  ,*-4
E1315       TF  FAC,FX1
E1316       TD  FAC-4,BLKN
E1317       CF  FAC-4
E1318       TF  -AVSLR,FAC
E1319       MA  -AVSLL,-AVSLR
E1320       TD  MES1+12,BLKN
E1330       WATYMES1
E1380LØØP   AM  99,10,10
E1390       AM  FAC,2,10
E1400       TF  -99,FAC
E1410       AM  99,10,10
E1430       TF  -99,FXZ
E1440       AM  -AVSLL,2,10
E1450       C   99,UPRLM
E1460       BNE LØØP
E1462       SM  99,10,10
E1464       TF  -99,FXZ
E1470       TF  BETA,-AVSLL
E1472       SF  BETA-3
E1473SAVE   DS  ,*
E1474       MM  BETA,5,10
E1480       TNF MES2+8,98
E1485       TF  SAVE,98
E1490       WATYMES2
E1500       SM  -AVSLL,2,10
E1501       MA  FAC,SAVE
E1505RETN   B7  1-ENT
E1525NEXT   H   2
E1999       B7  1-ENT
E2000****   EXTERNALLY CALLED UTILITIES
E2011*
E2012****   CALLING SEQUENCE = BTAM,SLADR,SLIP-ADDRESS,78 / RETURNS
E2013****       MACHINE ADDRESS IN 99 AFTER PØSSIBLY TURNING A DISK PAGE.
E2020       DAC 3,...
E2022SLADR  TF  99,LWRLM
E2025       BNC3NØTRP
            BNC1*+24
            H   4
```

```
                BNC2*+48
                PUT SWAP
                A   DISK+5,DISK+8
                BNC1*+24
                H   5
E2031NØTRP      BNF CHEK,SLADR-1
E2032           BB2
E2035CHEK       BNV ADD
E2036           CF  BACK+10
E2040ADD        A   98,SLADR-1
E2045           BV  ERR1
E2050           CM  SLADR-5,,1011
E2060PAGE       DS  ,*
E2070           BE  INCØR
E2080****       BLØCK NØT IN CØRE
E2290           H   1
E2400INCØR      C   99,UPRLM
E2410           BH  ERR1
E2411BACK       AM  *+8,,7910,TURN PØSSIBLE CHECK BACK ØN
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)
E3001MES2       DMES,A,XXXXX SLIP CELLS INITIALIZED.(E)
E3010NMS1       DMES,A,PSEUDØ-ADDRESS XXXX ØUT ØF RANGE(E)
E3120****       INTERNALLY REFERENCED CØNSTANTS
E3130LWRLM      DS  5,,ADDRESS ØF RIGHTMØST DIGIT ØF PSEUDØ-WØRD NUMBER (X0000)
E3132UPRLM      DS  5,,ADDRESS ØF  LAST PSEUDØ-WØRD (END ØF SECTØR BLØCK)
E3140DISK       DDA
E3145SWAP       DD  ,DISK
E3340AVSLL      DS  5,    , PØINTERS TØ FIELD ADDRESSES ØF RIGHT AND LEFT
E3350AVSLR      DS  ,LWRLM, LINKS ØF AVSL IN CØRE (SLIP CELL X0000).
E3440****       EXTERNALLY USED UTILITY ENTRY LINKAGES
E3450SLADR.NØP  SLADR
E3460           DØRG*-4
E3470           DC  1,'
E3680QQ         DAC 6,CØNT  ,
E3690           DVLC,5,CØNT
E3691           DAC 6,SETIND,
```

```
E3692       DVLC,5,SETIND
E3694       DAC 6,STRIND
E3696       DVLC,5,STRIND
E3700       DC  2,'
E3710LENGTHDS  ,QQ-1
E3720       DEND
```

```
ZZ    SETIND 0646

ZZSPS

*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
G1050      DSC  17,0,0
G1060      DØRGQ-100
G1070ID    DSA
G1080LNKL  DSA
G1090LNKR  DSA
G1100CELL  DSA
G1110      DC   1,'
G1112CALL  BTM  ,*+11
G1114INITASDS   ,*-5
G1120      DS   5
G1122ENT   B    CALL,,,TRANSFER VECTØR
G1123SLADR DS   ,*-5
G1130      TFM  TF+6,ID-4
G1140      AM   TF+6,4,10
G1150      AM   ENT-1,5,10
G1160      TF   CF+11,1-ENT
G1170      BNF  *+36,CF+11
G1180CF    CF   CF+11
G1190      TF   CF+11,-CF-11
G1200TF    TF   ,CF+11
G1210      AM   TF+6,1,10
G1220      BNR  ENT+24,-TF-6
G1230      AM   ENT-1,1,10
G2010      MA   *+35,-CELL
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 SUBPRØGRAM
G2050SETDIRDS   ,*-5
```

```
G2060          DSA -ID
G2070          DSA -LNKL
G2080          DSA -LNKR
G2090CELL.     DSA
G2180          B7  1-ENT
G2190QQ        DAC 6,SETDIR,
G2200          DVLC,5,SETDIR
G2202          DAC 6,INITAS,
G2204          DVLC,5,INITAS
G2210          DC  2,'
G2220LENGTHDS  ,QQ-1
G2230          DEND
```

```
ZZ    STRIND 0648

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*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
H1070DATUM DSA
H1080ICL   DSA
H1110      DC  1,'
H1112CALL  BTM ,*+11
H1114INITASDS  ,*-5
H1120      DS  5
H1125ENT   B   CALL,,,TRANSFER VECTØR
H1126SLADR DS  ,*-5
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
H1180CF    CF  CF+11
H1190      TF  CF+11,-CF-11
H1200TF    TF  ,CF+11
H1210      AM  TF+6,1,10
H1220      BNR ENT+24,-TF-6
H1230      AM  ENT-1,1,10
H2010      MA  *+35,-ICL
H2011      MF  *+23,-ICL
H2020      BTAM-SLADR,,78,CALL TØ INTERNAL UTILITY RØUTINE
H2030      TFL FAC,-DATUM
H2040      TFL -99,FAC
H2180      B7  1-ENT
H2190QQ    DAC 6,INITAS,
```

```
H2192       DVLC,5,INITAS
H2194       DC  2,'
H2196LENGTHDS  ,QQ-1
H2200       DEND
```

60592 1

```
ZZ     KGET    0672

*NAME KGET
*ID NUMBER 0672
I1030       DAC 6,KGET  ,7-Q
```

```
ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME KTEST
I1000****  KGET   0672
I1010Q     DS  ,*+101
I1020      DC  6,987898,5-Q
           DAC 6,KTEST ,7-Q
I1040      DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
I1050      DSC 17,0,0
I1060      DØRGQ-100
I1061FAC   DS  ,2492
I1062BUF   DAS 89,, MAY RELØCATE TØ FØRTRAN BUFFER AT LØC 2521
I1063BETA  DS  ,2630
I1064STZERØDS  ,2771
I1065FXZ   DS  ,3099
I1066****   =9, .=5, )=6, +=3, $=7, *=8, -=2, /=11, ,=13, (=10, ==1, '=14.
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
I1070INDV  DS  2,, 06 ØR 10
I1072      DGM
I1074NEXT  DS  ,PØINT-4, DTA ØR DCA
I1075MES   DMES,A, EXIT CALLED BY KGET(E)
I1078INPT  DSA
I1080NCØDE DSA
I1110      DC  1,'
I1120      DS  5
I1130ENT   TFM TF+6,INPT-4
I1140      AM  TF+6,4,10
I1150      AM  ENT-1,5,10
I1160      TF  CF+11,1-ENT
I1170      BNF *+36,CF+11
I1180CF    CF  CF+11
I1190      TF  CF+11,-CF-11
I1200TF    TF  ,CF+11
I1210      AM  TF+6,1,10
I1220      BNR ENT+12,-TF-6
I1230      AM  ENT-1,1,10
I1240      BSBA*+12
I1300      TF  FAC,FXZ
I1310      TF  BETA,FXZ
```

```
I1320        BNF REG,-INPT
I1330        TD  -PØINT,RKM
I1340        B7  ØUT
I1350REG     BNR NØRED,-PØINT
I1380        TFM PØINT,BUF
I1390        MA  DEVIC,-INPT
I1400        A   DEVIC,DEVIC
I1410        BD  TYPE,DEVIC
I1420        TFM INDV,10,10
I1430        B7  *+20
I1440TYPE    TFM INDV,06,10
             H   BUF-1
I1443****    CLEAR BUFFER -- IMAGE IS 87 INPUT CHARACTERS (BUF -- BUF+172),
I1444****        1 BLANK (BUF+174), AND 1 RECØRD MARK (BUF+176).
I1445RERE    TR  BUF-1,STZERØ-1
             H   BUF+86
I1450        TR  BUF+86,STZERØ
             H   BUF+174
I1460ALLBL   BD  *+24,INDV-1
I1462        RCTY
I1475        GET NEXT
I1476        BD  *+24,INDV-1
I1477        BC4 RERE
I148C        BLXM*+12,174(A1)
I148E        CM  BUF-2(A1),,10
I148G        BNE NBLK
I148J        BCXM*-24,-2(A1)
             H   6
I148L        B7  ALLBL
I148NNBLK    TDM BUF+2(A1)
I148PRKM     DC  1,',*
I148R        BNR *+24,BUF-2(A1)
I148T        TFM BUF-2(A1),22,10
I148V        BCXM*-24,-2(A1)
I148W        BNC3NØRED+12
             H   7
I149A        RCTY
I114C        WATYBUF
I114E        B7  NØRED+12
I114GNØRED   AM  PØINT,2,10
I114J        CM  -PØINT,22,10
I114L        BNE NØRKM
I1490        BNF ØKGØ,RMSW
```

```
I1492       WATYMES
I1500       CALLEXIT
I1525ØKGØ   SF   RMSW
I1530       B7   REG
I1540NØRKM  TF   FAC,-PØINT
I1550       CM   FAC,70,10
I1560       BL   *+32
I1565****   NUMERIC = 12
I1570       TFM  BETA,12,10
I1580       B7   DØNE
I1590       CM   FAC,41,10
I1600       BL   *+32
I1605****   ALPHABETIC = 04
I1670       TFM  BETA,04,10
I1680       B7   DØNE
I1685****   NEITHER -- USE TABLE LØØK-UP
I1700       BLX  *+12,FAC(A1)
I1710       BX   *+12,FAC(A1)
I1720       TF   BETA,TABLE(A1)
I2010DØNE   CF   FAC-1,,10
I2020DEVIC  DS   ,*
I2030RMSW   DS   ,*-2
I2040       CF   BETA-1
I2050       MF   BETA,RMSW
I2060       CF   RMSW
I2070ØUT    TF   -NCØDE,BETA
I2180       BSBB1-ENT
I2194LENGTHDC    1,'
I2200       DEND

$DEØF DEFINE END ØF S.P.S. PRØGRAM FILE -- D.P.K.
```

60594

1

```
ZZ    LCNTR  0637

ZZFØR

*FANDKØ810
*LDISKLCNTR 0637
      FUNCTIØN LCNTR (LST)                                    LCNT0010
      LCNTR=LNKR (CØNT (LST+1) )                              LCNT0020
      RETURN                                                  LCNT0030
      END                                                     LCNT0040
```

```
ZZ    LSTNAM 0639

ZZFØR

*FANDK0810
*LDISKLSTNAM0639
      FUNCTIØN LSTNAM (LST)                                          LSTN0010
      LSTNAM=LNKL (CØNT (LST+1) )                                    LSTN0020
      CALL SETDIR (0,LSTNAM,LSTNAM,LSTNAM)                           LSTN0030
      RETURN                                                         LSTN0040
      END                                                            LSTN0050
```

```
ZZ     INTLBL 0649

ZZFØR

*FANDK0810
*LDISKINTLBL0649
      FUNCTIØN INTLBL (I)                                          INTL0010
      INTLBL=I                                                     INTL0020
      RETURN                                                       INTL0030
      END                                                          INTL0040
```

```
ZZ    FLTLBL 0651

ZZFØR

*FANDK0810
*LDISKFLTLBL0651
      FUNCTIØN FLTLBL (X)                                          FLTL0010
      FLTLBL=X                                                     FLTL0020
      RETURN                                                       FLTL0030
      END                                                          FLTL0040
```

```
ZZ    NUCELL 0653

ZZFØR

*FANDK0810
*LDISKNUCELL0653
      FUNCTIØN NUCELL (X)                                          NUCE0010
      K=1                                                          NUCE0020
 10   M=LNKR(CØNT(-1))                                            NUCE0030
      IF(M)2,4,2                                                   NUCE0040
 4    GØ TØ (6,8),K                                                NUCE0050
 6    K=2                                                          NUCE0060
      CALL INITAS (X)                                              NUCE0070
      GØ TØ 10                                                     NUCE0080
 8    TYPE 1                                                       NUCE0090
 1    FØRMAT(18HNØ MØRE FREE SPACE)                                NUCE0100
      CALL EXIT                                                    NUCE0110
 2    IF(ID(CØNT(M))-1)12,14,12                                    NUCE0120
 14   CALL IRALST (CØNT(M+1))                                      NUCE0130
 12   CALL SETIND (-1,-1,LNKR(CØNT(M)),-1)                        NUCE0140
      CALL STRIND (0,M)                                            NUCE0150
      CALL STRIND (0,M+1)                                          NUCE0160
      NUCELL=M                                                     NUCE0170
      RETURN                                                       NUCE0180
      END                                                          NUCE0190
```

```
ZZ    RCELL  0655

ZZFØR

*FANDK0810
*LDISKRCELL 0655
      SUBRØUTINE RCELL (CELL)                                          RCEL0010
      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                                                              RCEL0060
```

```
ZZ    MTLIST 0657

ZZFØR

*FANDK0810
*LDISKMTLIST0657
      FUNCTIØN MTLIST (LST)                                          MTLI0010
      L=LØCT(LST)                                                    MTLI0020
      IF(LISTMT(L)) 2,4,2                                            MTLI0030
 2    LR=LNKR(CØNT(L))                                               MTLI0040
      LL=LNKL(CØNT(L))                                               MTLI0050
      CALL SETIND (-1,L,L,L)                                         MTLI0060
      CALL SETIND(-1,-1,LR,LNKL(CØNT(-1)))                           MTLI0070
      CALL SETIND(-1,LL,-1,-1)                                       MTLI0080
      CALL SETIND(-1,-1,0,LL)                                        MTLI0090
 4    MTLIST=L                                                       MTLI0100
      RETURN                                                         MTLI0110
      END                                                            MTLI0120
```

```
ZZ    IRALST 0659

ZZFØR

*FANDK0810
*LDISKIRALST0659
      FUNCTIØN IRALST (LST)                                          IRAL0010
      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
      IF(N)6,8,6                                                     IRAL0080
 6    NEW=NUCELL(X)                                                  IRAL0090
      CALL SETIND(1,-1,-1,NEW)                                       IRAL0100
      CALL STRIND (N,NEW)                                            IRAL0110
      CALL RCELL(NEW)                                                IRAL0120
 8    CALL RCELL(L)                                                  IRAL0130
 2    RETURN                                                         IRAL0140
      END                                                            IRAL0150
```

```
ZZ    LDUMP  0661

ZZFØR

*FANDK0810
*LDISKLDUMP 0661
      FUNCTIØN   LDUMP (N1,N2)                                              LDUM0010
      IT=2-N1+N1/2*2                                                        LDUM0012
C     IT.. 1=ØDD PØSITIVE, 2=EVEN, 3=ØDD NEGATIVE.                          LDUM0014
      LL=LNKL(CØNT(-1))                                                     LDUM0016
      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
     1       10H     AVSL=,216)                                             LDUM0031
      LDUMP=0                                                               LDUM0040
      GØ TØ (4,6,6),IT                                                      LDUM0042
 6    PUNCH 5                                                               LDUM0044
 5    FØRMAT(65H0   ADDRESS ØF FIRST CELL MUST BE ØDD AND PØSITIVE--LDUMLDUM0046
     1P ØMITTED)                                                            LDUM0047
      RETURN                                                                LDUM0050
 4    DØ 2 I=N1,N2,2                                                        LDUM0060
      LL=LNKL(CØNT(I))                                                      LDUM0080
      LR=LNKR(CØNT(I))                                                      LDUM0090
      IX=ID(CØNT(I))                                                        LDUM0100
      IW=INTLBL(CØNT(I+1))                                                  LDUM0110
      XW=CØNT(I+1)                                                          LDUM0120
      LDUMP=LDUMP+1                                                         LDUM0122
 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       6HDATUM=,I11,2X,2HØR,E16.10)                                   LDUM0150
      RETURN                                                                LDUM0160
      END                                                                   LDUM0200
```

```
ZZ     IQUAL  0665

ZZFØR

*FANDK0810
*LDISKIQUAL 0665
      FUNCTIØN IQUAL (K,L)                                          IQUA0010
      IF(LNKL(K)-LNKL(L))2,4,2                                      IQUA0020
 4    IF(LNKR(K)-LNKR(L))2,6,2                                      IQUA0030
 6    IF(ID(K)-ID(L))2,8,2                                          IQUA0040
C8    IQUAL=K-L                                                     IQUA0050
 8    IQUAL=0                                                       IQUA0050
      RETURN                                                        IQUA0060
 2    IQUAL=1                                                       IQUA0070
      RETURN                                                        IQUA0080
      END                                                           IQUA0090
```

```
ZZ    LISTMT 0667

ZZFØR

*FANDK0810
*LDISKLISTMT0667
      FUNCTIØN LISTMT (LST)                                          LIST0010
      L=LØCT(LST)                                                    LIST0020
      LISTMT=IQUAL(CØNT(L),CØNT(LNKR(CØNT(L))))                      LIST0030
      RETURN                                                         LIST0040
      END                                                            LIST0050
```

```
ZZ    NAMTST 0669

ZZFØR

*FANDK0810
*LDISKNAMTST0669
      FUNCTIØN NAMTST (L)                                                    NAMT0010
      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
      RETURN                                                                 NAMT0060
 2    NAMTST=1                                                               NAMT0070
      RETURN                                                                 NAMT0080
      END                                                                    NAMT0090
```

```
ZZ    LØCT   0671

ZZFØR

*FANDK0810
*LDISKLØCT  0671
      FUNCTIØN LØCT (LST)                                          LØCT0010
      IF(NAMTST(LST))2,4,2                                         LØCT0020
 4    LØCT=LST                                                     LØCT0030
      RETURN                                                       LØCT0040
 2    TYPE 1                                                       LØCT0050
 1    FØRMAT(16HØPERAND NØT LIST)                                  LØCT0060
      CALL EXIT                                                    LØCT0070
      END                                                          LØCT0080

$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE -- D.P.K.
```

60600 T

```
ZZ     LIST   0675

ZZFØR

*FANDKØ810
*LDISKLIST  0675
      FUNCTIØN 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
```

```
ZZ     TØP     0677

ZZFØR

*FANDK0810
*LDISKTØP    0677
       FUNCTIØN TØP(P)
       TØP=CØNT(LNKR(CØNT(LØCT(P)))+1)
       RETURN
       END
```

```
ZZ     BØT     0679

ZZFØR

*FANDK0810
*LDISKBØT    0679
       FUNCTIØN BØT(P)
       BØT=CØNT(LNKL(CØNT(LØCT(P)))+1)
       RETURN
       END
```

60602

1

```
ZZ     PØPTØP 0681

ZZFØR

*FANDK0810
*LDISKPØPTØP0681
       FUNCTIØN PØPTØP(P)
       PØPTØP=DELETE(LNKR(CØNT(LØCT(P))))
       RETURN
       END
```

```
ZZ     PØPBØT 0683

ZZFØR

*FANDK0810
*LDISKPØPBØT0683
       FUNCTIØN PØPBØT(P)
       PØPBØT=DELETE(LNKL(CØNT(LØCT(P))))
       RETURN
       END
```

60603

1

```
ZZ     NEWTØP 0685

ZZFØR

*FANDK0810
*LDISKNEWTØP0685
       FUNCTIØN NEWTØP(P,Q)
       NEWTØP=NXTRGT(P,LØCT(Q))
       RETURN
       END
```

```
ZZ     NEWBØT 0687

ZZFØR

*FANDK0810
*LDISKNEWBØT0687
       FUNCTIØN NEWBØT(P,Q)
       NEWBØT=NXTLFT(P,LØCT(Q))
       RETURN
       END
```

```
ZZ    NXTRGT 0689

ZZFØR

*FANDK0810
*LDISKNXTRGT0689
      FUNCTIØN NXTRGT(M,A)
      IR=NUCELL(Z)
      NXTRGT=IR
      LR=LNKR(CØNT(A))
      CALL SETIND(-1,IR,-1,LR)
      CALL SETIND(-1,-1,IR,A)
      CALL SETIND(0,A,LR,IR)
      IF(NAMTST(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
      END
```

```
ZZ    NXTLFT 0691

ZZFØR

*FANDK0810
*LDISKNXTLFT0691
      FUNCTIØN NXTLFT(M,A)
      IL=NUCELL(Z)
      NXTLFT=IL
      LL=LNKL(CØNT(A))
      CALL SETIND(-1,-1,IL,LL)
      CALL SETIND(-1,IL,-1,A)
      CALL SETIND(0,LL,A,IL)
      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)
      RETURN
      END
```

```
ZZ    LISTRD 0695

ZZFØR

*FANDK0810
*LDISKLISTRD0695
      FUNCTIØN LISTRD(LST,IN)
      EQUIVALENCE (NUM,WØRD)
      CALL KGET(-1)
      ISUB=1
   10 ICHAR=KGET(IN,ICØDE)
      IF(ICØDE-10)10,20,10
   20 LISTRD=LST
      CALL NEWTØP(LST,LIST(STACK))
   40 IDELSW=0
      NUM=0
      IFLTSW=0
      ISINSW=0
   50 ICHAR=KGET(IN,ICØDE)
      IF (ICØDE) 51,51,54
C END ØF LIST
   51 IF(IDELSW)53,53,52
   52 CALL NEWBØT(WØRD,TØP(STACK))
   53 CALL IRALST(STACK)
      RETURN
   54 GØ TØ (110,120,130,140,150,160,170,180,190,200,210,220,230),ICØDE
C EQUAL SIGN
  110 IF(IDELSW)112,112,111
  111 CALL NEWBØT(WØRD,TØP(STACK))
  112 NUM=0
      CALL KPUT(ICHAR,WØRD,TØP(STACK))
      CALL NEWBØT(WØRD,TØP(STACK))
      GØ TØ 40
C MINUS SIGN
  120 IF(IDELSW)121,121,111
  121 IDELSW=1
      ISINSW=-1
      CALL KPUT(ICHAR,WØRD,TØP(STACK))
      GØ TØ 50
C PLUS SIGN
  130 IF (IDELSW)131,131,111
  131 IDELSW=1
      ISINSW=1
```

```
          CALL KPUT(ICHAR,WØRD,TØP(STACK))
          GØ TØ 50
C LETTER
  140 IDELSW=1
          CALL KPUT(ICHAR,WØRD,TØP(STACK))
          GØ TØ 50
C DECIMAL PØINT
  150 IF(IDELSW)112,112,152
  152 IF(IDELSW)153,111,111
  153 WØRD=NUM
          SIGN=ISIGN
          IFLTSW=1
          DEC=10.
          GØ TØ 50
C RIGHT PAREN
  160 IF(IDELSW)162,162,161
  161 CALL NEWBØT(WØRD,TØP(STACK))
  162 CALL PØPTØP(STACK)
          ISUB=ISUB-1
          IF(ISUB)163,163,40
  163 CALL IRALST(STACK)
          RETURN
C DØLLAR SIGN
  170 GØ TØ 110
C ASTERIK
  180 GØ TØ 110
C BLANK
  190 IF(IDELSW)50,50,191
  191 CALL NEWBØT(WØRD,TØP(STACK))
          GØ TØ 40
C LEFT PAREN
  200 IF(IDELSW)202,202,201
  201 CALL NEWBØT(WØRD,TØP(STACK))
  202 ISUB=ISUB+1
          CALL NEWBØT(LIST(9),TØP(STACK))
          CALL NEWTØP(BØT(TØP(STACK)),STACK)
          GØ TØ 40
C SLASH
  210 GØ TØ 110
C NUMBER
  220 IF(ISINSW)223,221,223
  221 IF(IDELSW)222,222,224
  222 ISINSW=1
  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
      WØRD=WØRD+SIGN*VAL/DEC
      DEC=DEC*10.
      GØ TØ 50
C CØMMA
  230 GØ TØ 110
      END
```

```
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)
      LL=LNKL(CØNT(K))
      LR=LNKR(CØNT(K))
      CALL RCELL(K)
      CALL SETIND(-1,-1,LR,LL)
      CALL SETIND(-1,LL,-1,LR)
      RETURN
      END
```

```
ZZFØR

ZZ     SEQRDR 0697

*FANDK0810
*LDISKSEQRDR0697
       FUNCTIØN SEQRDR(LST)
       SEQRDR=CØNT(LØCT(LST))
       RETURN
       END
```

```
ZZ    SEQLR  0699

ZZFØR

*FANDK0810
*LDISKSEQLR 0699
      FUNCTIØN SEQLR(Z,N)
      L=LNKR(Z)
      Z=CØNT(L)
      SEQLR=CØNT(L+1)
      N=ID(Z)-1
      RETURN
      END
```

```
ZZ    SEQLL  0701

ZZFØR

*FANDK0810
*LDISKSEQLL 0701
      FUNCTIØN SEQLL(Z,N)
      L=LNKL(Z)
      Z=CØNT(L)
      SEQLL=CØNT(L+1)
      N=ID(Z)-1
      RETURN
      END
```

```
ZZ    SEQSR  0703

ZZFØR

*FANDK0810
*LDISKSEQSR 0703
      FUNCTIØN SEQSR(Z,N)
      IF(ID(Z)-1)4,5,4
    5 L=LNKR(CØNT(CØNT(LNKL(CØNT(LNKR(Z)))+1)))
      GØ TØ 3
    4 L=LNKR(Z)
    3 IF(ID(CØNT(L))-1)1,2,1
    1 SEQSR=CØNT(L+1)
      Z=CØNT(L)
      N=ID(Z)-1
      RETURN
    2 L=LNKR(CØNT(CØNT(L+1)))
      GØ TØ 3
      END
```

```
ZZ    SEQSL  0705

ZZFØR

*FANDK0810
*LDISKSEQSL 0705
      FUNCTIØN SEQSL(Z,N)
      IF(ID(Z)-1)4,5,4
    5 L=LNKL(CØNT(CØNT(LNKL(CØNT(LNKR(Z)))+1)))
      GØ TØ 3
    4 L=LNKL(Z)
    3 IF(ID(CØNT(L))-1)1,2,1
    1 SEQSL=CØNT(L+1)
      Z=CØNT(L)
      N=ID(Z)-1
      RETURN
    2 L=LNKL(CØNT(CØNT(L+1)))
      GØ TØ 3
      END

$DEØF DEFINE END ØF FØRTRAN PRØGRAM FILE -- R.L.S.
```

```
ZZJØB                                   29270126507387+215

ZZJØB                                   29270126507387+215

ZZ    KPUT   0694

ZZSPS

*LIST CARD
*ASSEMBLE RELØCATABLE
*STØRE RELØADABLE
*NAME KPUT
*ID NUMBER 0694
     ***** KPUT   0694
     Q      DS   ,*+101
            DC   6,987898,5-Q
            DAC  6,KPUT  ,7-Q
            DVLC22-Q,5,LENGTH,2,8,2,10,5,ENT-6,5,0,30,0
            DSC  17,0,0
            DØRGQ-100
     FAC    DS   ,2492
     FXZ    DS   ,3099
     BETA   DS   ,2630
     IPNT   DS   5
     ICHAR  DSA
     WØRD   DSA
     LST    DSA
            DC   1,'
            DS   5
     ENT    TFM  TF+6,ICHAR-4
            AM   TF+6,4,10
            AM   ENT-1,5,10
            TF   CF+11,1-ENT
            BNF  *+36,CF+11
     CF     CF   CF+11
            TF   CF+11,-CF-11
     TF     TF   ,CF+11
            AM   TF+6,1,10
            BNR  ENT+12,-TF-6
            AM   ENT-1,2,10
            BSBA*+12
            CM   -WØRD,,10
            BNE  *+24
            TFM  IPNT
```

```
          AM   IPNT,2,10
          CM   IPNT,12,10
          BNE NØT
          BTM ,*+11
NEWBØTDS  ,*-5
          DSA -WØRD
          DSA -LST
          TF  -WØRD,FXZ
          TFM IPNT,2,10
NØT       BLX *+12,IPNT(A1)
          TF  FAC,-WØRD
          TF  BETA,-ICHAR
          TD  FAC-11(A1),BETA-1
          TD  FAC-10(A1),BETA
          TF  -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.
```

```
ZZFØRX

*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 )ZZZZZZZZZ
(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)
```

60611

1

ZZZZ END ØF JØB

SLIP | 2052 1963-67 OK

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 G. 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

October 26, 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 intend 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,

Phyllis Fox

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/ls.

ASSOCIATION FOR COMPUTING MACHINERY 211 EAST 43 STREET NEW YORK 17, NEW YORK

COPY

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,

llg.

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 ~~by~~ 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

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 aware of.

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

Very truly yours,

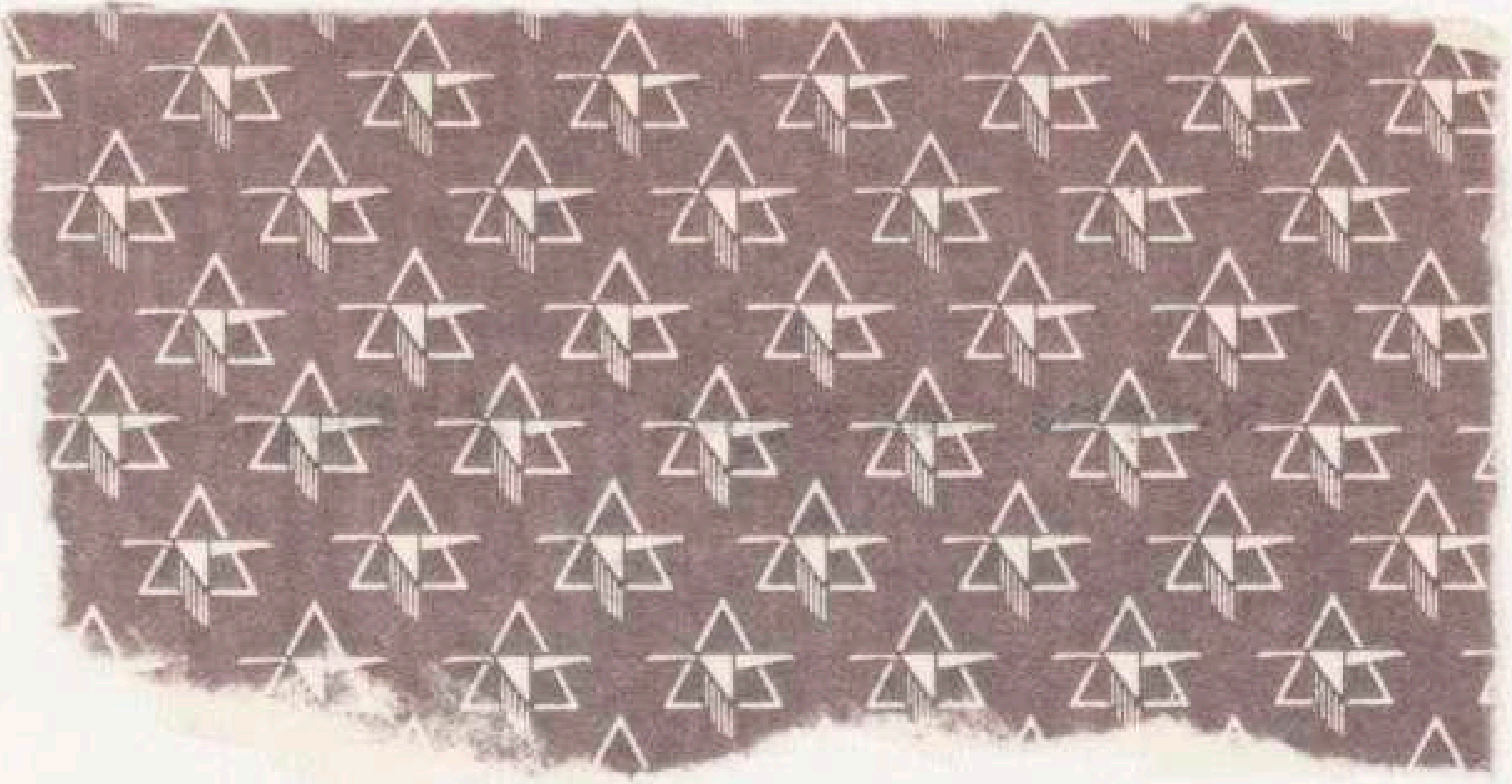
David L. Parnas

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

12214 LAKEWOOD BOULEVARD, DOWNEY, CALIFORNIA 90241

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

10th March 1966
PGR/BS

Prof. J.Weizenbaum,
Department of Electrical Engineering,
Massachusetts Institute of Technology,
Cambridge,
Mass,
U.S.A.

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.

BY AIR MAIL
PAR AVION
AIR LETTER
AEROGRAMME

POST PAID 0/6 GREAT BRITAIN NE-924

MAC
R(o)
1.5.809

Prof. J.Weizenbaum,
Department of Electrical Engineering,
Massachusetts Institute of Technology,
Cambridge,
MASS., U.S.A.

First fold here

Second fold here

Sender's name and address: P.G.Raymont,
Director of Computing Laboraoty,
Leicester Regional Dollege of Technology,
Leicester, England.

AN AIR LETTER SHOULD NOT CONTAIN ANY ENCLOSURE:
IF IT DOES IT WILL BE SURCHARGED
OR SENT BY ORDINARY MAIL.

Form approved by the Postmaster General, No. 71995/2E

To open cut here

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

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 N. Lieberman

Robert N. Lieberman
Research Graduate
Assistant

RNL:ags

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

211 EAST 43 STREET NEW YORK 17, NEW YORK 212 YU 6-3055

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.

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,

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,

Paul M. Kjeldergaard

Paul M. Kjeldergaard
Research Associate

/jnm

THE BOEING COMPANY

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.

2. 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

(Mrs.) Masayo Kumakura

for 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 answers to 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 I have about 6 more months of work for completing my Ph.D. in linguistics.

Sincerely,

Dick V.

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 INTEGER 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 A. White

Richard A. White

RAW:nam

Résidence : 731-6004.

Bureau : 737-0862.

PROVINCE de MONTRÉAL

Les Frères de Saint-Gabriel

ADMINISTRATION PROVINCIALE
5638, rue Canterbury
Montréal 26.

November 29, 1964

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,

Ernest Faubert

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

David 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 Communications. 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,

John Bossons

JB/jlb

SUN OIL COMPANY
1608 WALNUT STREET
PHILADELPHIA 3, PA.

C. G. KIRKBRIDE
VICE PRESIDENT - RESEARCH & ENGINEERING

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

November 18, 1964

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,

David R. Bamberger

David R. Bamberger

lap

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,

Roger W. Elliott

Roger W. Elliott