** ESTIMATE OF THE NUMBER OF FACTORS. LOG TO THE BASE 2 OF N IS **
** THE ESTIMATE (CLEARLY AT LEAST THIS MANY IS REQUIRED FOR N A **
** POWER OF 2), THE NEED TO ESTIMATE THE NUMBER OF FACTORS IS **
** DUE TO PL/I'S REQUIREMENT ON ARRAY DECLARATIONS. THIS IS AN **
** ANNOYING FEATURE OF A PL/I BASED THEORY WHICH WOULD NOT APPEAR **
** IN, SAY, A LISP BASED THEORY.

** /* ASSUME **
03: DOM(A,L) & DOM(A,U) & U-L >= LOG(2,N) & N > 1, AA = A; **
** /* ATTAIN **
03: AT1: H = PRD(A,L,M) & L<=M<=U, DOM(A,H), **
03: AT2: ALL I FIXED WHERE L<=I<=M, (PRD(A,L,M) & DIV(A,L,N)); **
03: AT3: ALL (I,J) FIXED WHERE L<=I<=M, A(I)<A(J). **
03: AT4: ALL I FIXED WHERE LBOUND(A) <= I <= UBOUND(A) <= I = A(I); **
** /* ARBITRARY D FIXED WHERE U-L < D; **
03: DECLARE(j,k) FIXED; **
03: DECLARE AAA(BOUND(A)), BBOUND(A), I) FIXED; **
03: DECLARE ABOUND(A), I) FIXED; **
03: DECLARE SOME D FIXED, (D=0 & N=D) BY INTRO,N; **
03: DECLARE LOG(N),>=0 BY FUNCTION,LOG(2,N); **
** /* FIND THE LEAST PRIME FACTOR OF N, CALL IT P **
03: /* P = LEAST PRIME FACTOR(N); **
03: /* /* PRIMEP AND DIVP(N) & ALL I FIXED WHERE 1 <= P < DIVP(1,N) **
03: /* BY FUNCTION, LEAST PRIME FACTOR(N); **
03: /* /* DIVIDE OUT THE LEAST PRIME FACTOR AND MAKE IT THE FIRST **
03: /* FACTOR OF THE PRODUCT, A(L). IF N IS COMPLETELY FACTORED **
03: /* AS A RESULT, THEN STOP, OTHERWISE FACTOR N/P IN THE SAME WAY. **
03: /* /* P=0 BY ARITH, P=1; **
03: /* /* MOD(N,P) = 0 BY ALLEL, DIVMOD_EQUIVALENT, N,P; **
03: /* /* ATTAIN N/P=1 & AA = A; **
03: IF N/P = 1 THEN **
03: DO; **
03: /* N IS COMPLETELY FACTORED, ALL THE REQUIRED PROPERTIES **
03: /* ALL AT2, AT3, CAN BE PROVED TRIVIALLY FROM THE INFORMATION **
03: /* THAT P IS PRIME, P DIVIDES N, N/P=1, AND M=1. **
03: /* FOR THE ARRAY RULE SAVE A IN AR **
03: /* /* AR = A; **
03: /* /* AR = A; /* AR = AA; **
03: /* N = L1; /* L = 2; **
03: M = L1; **
03: A(L1) = P; **
03: DOM(A,M); **
03: SOME D FIXED, (D=0 & N=0) BY INTRO,N; **
03: SOME D FIXED, (D=0 & NONS(U+1,L), D) BY FUNCTION, **
03: PROD_TERMINATION(A(1,M)); **
03: PROD(A,L,M) = A(L) BY FUNCTION, PROD(A,L,M); **
03: N = (N/P)*P + MOD(N,P) BY FUNCTION, DIVISION(N,P); **
03: N = 1+P; **

/**** THIS IS THE FILE DIVN12P.LGO, ADVANCED DIVISION THEORY *****/

** PRIME_FACTORIZATION: PROCEDURE((N,A,L,M) /* = AA */); **
DECLARE (A(*),M) FIXED; /* READWRTS PARAMETERS **
DECLARE (N,L,U) FIXED /* READONLY */; /* DECLARE (AA(*)) FIXED READONLY; **
*/ FACTOR AN INTEGER GREATER THAN 1 INTO A PRODUCT **
*/ OF PRIMES, PRD(A,L,M). THE PRIMES APPEAR IN NONDECREASING **
*/ ORDER, A(I)<A(J) FOR I<=J. M IS THE NUMBER OF FACTORS. **
*/ OBS WHILE U IS THE UPPER BOUND OF THE ARRAY A WHICH IS AN **

X/Y > 0 BY FUNCTION, DIVISION(X,Y); **
(Y/X) > 0 BY ARITH, X/Y > 0, Y > 0; **
*/ SET UP CONDITIONS FOR MONOTONEGENCY LEMMAS */
** /* EQUATION */
EQ1: X = (X/Y)*Y + MOD(X,Y) BY FUNCTION, DIVISION(X,Y); **
EQ2: MOD(X,Y) > 0 BY FUNCTION, MOD(X,Y); **
** EQUATION */
IEQ0: X = (X/Y)*Y BY ARITH, EQ1 - EQ2; **
** // SET UP CONDITIONS FOR MONOTONEGENCY LEMMAS */
** /* TERMINATION OF LOG */
** /* SOME D FIXED, (D=0 & Y=0) BY INTRO,Y; **
** /* SOME D FIXED, (D=0 & X-Y=0) BY INTRO,X/Y; **
** L1: LOG(X,Y) > 0 BY FUNCTION, LOG(X,Y); **
** L2: LOG(X,Y) >= 0 BY FUNCTION, LOG(X,Y); **
** /* LOG(B,X/Y) + LOG(B,Y) >= 0 BY ARITH, X/Y + Y, 11/2; **
** /* LOG(B,X/Y) <= LOG(B,Y) >= 0 BY ALLEL, EXP_POSITIVE, B, LOG(B,X/Y); **
** /* B = LOG(B,X/Y) <= X/Y BY FUNCTION, LOG(B,X/Y); **
** /* B = LOG(B,X/Y) <= Y BY FUNCTION, LOG(B,Y); **
** /* B = LOG(B,X/Y) + LOG(B,Y) > 0 BY ALLEL, EXP_POSITIVE, B, LOG(B,X/Y), LOG(B,Y); **
** /* /* IN ORDER TO COMPARE ARGUMENTS */
** /* NOW MULTIPLE IEQ1 BY IEQ2 IN STAGES */
L1: LOG(X,Y) >= LOG(B,Y) <= (X/Y) * LOG(B,Y); **
** BY ARITH, IEQ1, *, X/Y; **
** /* LOG(B,X/Y) >= (X/Y) * LOG(B,Y); **
** /* /* /* SUBST IN EQ3 */
** /* NOW TAKE LOGARITHMS AND COMPARE */
** /* LOG(B,X/Y) + LOG(B,Y) <= LOG(B,X/Y); **
** /* BY ALLEL, LOG_MONOTONE, B, LOG(B,X/Y), LOG(B,Y); **
** /* LOG(B,X/Y) + LOG(B,Y) <= LOG(B,X/Y); **
** /* BY ARITH, LOG, LOG, LOG; **
** /* F1: LOG(B,X/Y) + LOG(B,Y) <= LOG(B,X/Y); **
** /* F2: LOG(B,X/Y) + LOG(B,Y) <= LOG(B,X/Y); **
** /* / FROM IEQ0 */
** /** CONCLUSION */
** /* / FROM IEQ0 */
** BY ARITH, F1, F2; **

qed;

/*
N = P = PROD(A[1], N);  
M <= U;  
ALL I FIXED WHERE L <= I <= M, (PRIME(A[1])) DIV(A[1], N)) BY INTO;  
PROOF:  
I = L BY ARITH, L <= I <= M, M = M;  
A(I) = P;  
QED:  
ALL (i, j) FIXED WHERE L <= I <= J <= M, A(I) <= A(J) BY INTO.  
PROOF:  
I = L BY ARITH, L <= I <= J <= M, M = M;  
J = L BY ARITH, L <= I <= J <= M, M = M;  
A(I) <= A(J) BY ARITH, A(1) = A(J)  
QED;  
ALL I FIXED WHERE LBOUND(A(1)) <= I <= L, A(I) = A(I) BY INTO,  
PROOF:  
I = L BY ARITH, L <= I <= L, M = M;  
A(I) = A(I);  
A(I) = AA(I);  
QED;  
/* USE AR TO NAME A BEFORE ASSIGNMENT A(I) = P */  
*/ AR = A;  
*/ AR = A;  
*/ AR = A;  
N2 = N / P;  
N2 >= 1 BY ARITH, N2 <= N, M = M;  
N2 = N2;  
U - (N2/N) >= 1 BY ARITH, U - (N2/N) >= 1;  
LBOUND(A(1)) <= I <= L, L = L;  
DOM(A(1)) =  
QED;  
SHOW N/P IS GREATER THAN 1  
P <= N BY ARITH, LBOUND SIZE P, N;  
N/P >= 1 BY ARITH, LBOUND LEAD, N, P;  
SHOW U - (N2/N) >= 1 FOR TERMINATION  
U - (N2/N) >= 1 BY ARITH, U - (N2/N) >= 1;  
*/  
CALL PRIME_FACTORIZATION(N2, A1, L, N, M, A2, N, M) =  
IN LIST CONSEQUENCES  
/*  
C1: N2 = PROD(A[1], 1, M);  
C2: L1 <= N <= U;  
C3: ALL I FIXED WHERE L1 <= I <= N, (PRIME(I) + DIV(I, N2));  
C4: ALL (I, J) FIXED WHERE L1 <= I <= J <= N, (A(I) < A(J));  
C5: /* TRANSFER CONDITION ALLOWING PROOF THAT A(I) IS UNCHANGED  
ALL I FIXED WHERE LBOUND(A(1)) <= I <= L, A(I) = A(I);  
LBOUND(A, 1) = LBOUND(AAA, 1);  
DOM(AAA, L1) = DOM(AAA, U);  
CALL PRIME_FACTORIZATION(AA, L, N, M) =  
*/  
UL = UL + MODULUS(M+1, L) = U;  
N = A(N, P) + NMOD(N, P) BY FUNCTION, DIVISION(N, P);  
N = A(N, P) + NMOD(N, P) = A(N, P);  
N = A(N, M);  
SOME D FIXED, (D = 0 & MODULUS(M+1, L) = 0) BY FUNCTION,  
PROD_TERMINATION(A, L, M);  
/* */  

**QED:** /\ END OF BOUNDED ALL INTRO
/\ FINALLY PROVE THE TRANSFER CONDITION, AT4
/\ BY RELATING A BEFORE A(P) & P AND CALL OF
/\ PRIME_FACTORIZATION TO A AFTER.
/\ ALL I FIXED WHERE LBOUND(A(I))<2<z< u.< AA(A(I))=A(I)
/\ BY INTRO,PROOF;
/\ I<=l+1 BY ARITH,I<l; I=aL BY ARITH,I<l;
/\ AA(A(I))=A(I) BY ALLEL,C5,I;
/\ AA(A(I))=AA(A(I)) BY ALLEL,AA,AA_AAA,I;

**QED:**

RETURN:

**END PRIME_FACTORIZATION;**

**THEOREM**

/\ FOR (N,F(*),L,U) FIXED DEFINE PRIME_FACTORS(N,F,L,U) =
/\ ALL I FIXED WHERE L<=I<=u.(PRIME(F(I)) & DIV(F(I),N));
/\ FOR (F(*),L,U) FIXED DEFINE ORDERED(F,L,u) =
/\ ALL (I,J) FIXED WHERE L<=I<=J<=u. F(I)<=F(J);
/\ FOR (N,F(*),L,U) FIXED DEFINE FACTORIZATION(N,F,L,U) =
/\ N = PROD(F,L,u) & L<=u & PRIME_FACTORS(N,F,L,U) & ORDERED(F,L,u);

/*
** FTA_LEMMA: PROCEDURE(N,F,L,U) RETURNS(BIT(I));
DECLARE (N,F,L,U) FIXED;
/\ A LEMMA NEEDED TO SHOW THAT PRIME FACTORIZATION IS UNIQUE
/\ ASSUME N > 1. PRIME(F(I)) & DIV(F(I),N).
/\ DOMAIN: DOM(F,L,u) & DOM(F,U);
/\ FACT: FACTORIZATION(N,F,L,U);
/\ LESS: ALL I FIXED WHERE L<=I<=U. P < F(I);
/\ DEFINE FALSE = 0;B;
/\ ATTAIN FALSE;
/\ SHOW THAT SINCE P DIVIDES N AND IS PRIME, IT MUST DIVIDE
/\ ONE OF THE F(I) (A CONSEQUENCE OF PRIME_DIVIDES_LONG_PRODUCT);
/\ SINCE F(I) IS ALSO PRIME, P MUST EQUAL F(I). BUT THIS IS
/\ IMPOSSIBLE SINCE P<F(I).
/\ /*
/\ /* BODY OF THE PROOF ****
/\ L=U;
/\ DIV(F,P,PROD(F,L,U)); /* FROM FACTORIZATION AND DIV(F,P,U);
/\ ALL I FIXED WHERE L<=I<=U.F(I)>0 BY INTRO,
/\ PROOF;
/\ L<=u; /* DEFINITION OF DOM(F,I) AND DOMAIN ASSUME
/\ PRIME(F(I)) BY ALLEL,PRIME_FACTORS(N,F,L,U),I;
/\ F(I)>0 BY ARITH,F(I)>1; /* FROM DEF OF PRIME
/\ QED;
/\ DIV(F,P,PROD(F,L,U));
/\ SYMT2 thượng;
/\ BY ALLEL,PRIME_DIVIDES_LONG_PRODUCT,UP,L;
/\ POOR STYLE IN WRITING THE QUANTIFIERS IN
/\ PRIME_DIVIDES_LONG_PRODUCT RESULTS IN
/\ THIS FUSS TO GET AT THE STATEMENT;
/\ WE NEED TO BE CAREFUL ABOUT CAPTURING U-L;
/\ SO WE REMAKE THE BOUND VARIABLES OF SYMT2(U-L);
/\ LA: ALL (A(*),L,U) FIXED WHERE LL<=U & DOM(A,L,L) &