FORTRAN source program calculating the Goldbach-Xu's numbers with recursive method

Wan-Dong Xu

School of Science, Tianjin University, Tianjin 300072, China

E-mail : **xwandong@sohu.com**

Abstract: In this paper we have presented two source programs with FORTRAN 90 language to calculate the distribution of prime numbers in the sequence of odd numbers and the Goldbach-Xu's numbers for every even number respectively. Form the result we can know that this number is oscillatingly increased as an even number increases.

Keywords: Goldbach's conjecture; prime; distribution of primes; Goldbach-Xu's number

MSC: 11P32; 11A41; 11N05; 11N35

1. Introduction

Chinese famous mathematician Luo-Geng Hua had presented in 1979 a route to analyzingly deduce the solution numbers of Goldbach's problem for every even number, $r(N)$, through solving a indefinite equation, $ax+by=N$, here N is an even, $a>0$, $b>0$, and $(a, b)=1$ [1]. He was failed. After that time, some researchers devoted themselves in this route, and were also not succeeded [2]. Recently, we have advanced a recursive method to calculate the number of rest sum formulae of two unequal odd primes, which are to express every even number >6 in natural sequence [3]. Now we will list some programs of calculating them. By the results of calculating for them, we can know that the number of rest sum formulae, or say, Goldbach-Xu's number, is oscillatingly increased as an even number increases such that the Goldbach's conjecture is shown to be true.

2. Calculating the distribution of primes in the sequence of odd numbers

We have written a program, named PRIMECAL, with FORTRAN 90 language to calculate the distribution of odd prime numbers in the sequence of odd numbers in Appendix A. There is a input number in that program is NUPTO, it means to calculate the primes up to NUPTO, and there is a output file named D0101T1.DAT, in which an odd prime number is denoted by symbol "1" and an odd composite number by symbol "0", and in which there are 100 figures in every row to indicate 100 odd numbers in order. This is completely the same as ref. [4].

In the Table 1 we listed a distribution table for some prime numbers up to 12,600.

3. Calculating the Goldbach-Xu's numbers

There is a program calculating with a recursive method, named GOXUNUMB in Appendix B, with FORTRAN 90 language, to calculate the numbers of rest sum formulae of two unequal odd primes, or say, the Goldbach-Xu's numbers, for every even number. There is an input file named D0101T1.DAT, which is the output file in the section **2** above. And there is an output file named GOLDXUN1.DAT, which could be opened by the write-board in Windows XP. If one is first run this program the "status" in program should be set "new", after then, the "status" should be reset "old". In that program most of variant names are the same as symbols in ref. [3].

In the Table 2 we listed some Goldbach-Xu's numbers for every even numbers starting at 8 and ending up to 3366. And we can know that the Goldbach-Xu's number is oscillatingly increased as an even number increases such that the Goldbach's conjecture can be verified to be true.

There are two source programs in the author's hands, which could be sent readers if they need them and connect to the author.

Table1. The irregular distribution of primes in the sequence of odd numbers

0 1 2 3 4 5 6 7 8 9 1
0111011011010011001011010010011001011001010010001011011010000001010011000011001001010010011000011011
0000010000010110100110000100100100110010110000100000010110100000010010000110100100010010010100100010
1000100001100001100101001000101101000001000101000101001000001100000000100100001001001100100001001001
1001001011000000100001101001001100000101001000100001000100001000100100101000100101000101000000100001 0
0000110000110110000100000010110100000001011010000000000101000100001000101001001000000101001001000100 10
0000101001100001100100001100001100100000000101101001001000100100100000000000110000100010000100100100 0
1000001010010011001000001000010000000011010011001011010000011001000000000000000001001001000100000000 1
0000100000010110100100010110010000010000110110100100000100000100010000010010100100010100010100000010
1001101001100100001000000000100101100000000000010110000100000110000100010010010010000000001001011000 00
1000010000010001000000010000001001011011000010000010010010000000011000000011000000000010010001001011
0100010010000110000100000001000010010000011011000010000011000000011001011000010001000000000100000000 00
0101001000100000001101000100000001101000100100101000001100000000001001100101001000000100101100101001
0000010010010000000101001000001000100101000000000000100000000100001000101001100100000000000100000110 00
0000100010100000010000000100001101000100100101101001000101100100001100001000101000000010000100000110 010
1100000001000000010100100010010100000000010001000010010010001000010000010000000101001001100000000000 001
1000010001010000000101000100000100100000101001000000000100001100000001000000000000101100000100101000 0
0100100010100010000000000110110000010000000000000110010010010100110000010100000110000110000000110000
0001001000000000010000000100010110110000000000100010000010010000110100110010000110000010000110000100 0
0001001010010001001001000000001000000110100000001001010001000010001001001000000000001001100001000000 1010
0100000000110000100000010110000100000010100010000000001010011010011000001010000000001000000000010000 0
1101001001100100000000001100100000000100100000110010000010000000110100100000010110000000001000000000 00
1000010011000011000011000010011000011000010010001000000000000000100001100001000100100001000000000100 10
0000100000110000000010010100100100000000011000010000000100101101000000000000110000010010000000010001 0010
0100000000100000000110100110010010000100100000010000010000000011001010000000001000100000000000101101 001
1000001010000000100000000000000100001001000000100000010010000100000110100100010010000110100000001001 001
0100110000110000000010000010001000000000101001000000110010010010000000000000010010000000101000100001 00010
0000100000000101101000000000000100000100110000000100100100000010000100000010100000000000000001001001 001
0001001011000000100101100100000000000100110100000000011010000001100101000000000000100100101000100001 0000
0000000000100000001100101101100001000000100101000100001001000000000101100100000000000000010100010000 0
1001001000100100000101001100101001100001100000000100100000000001010000010000001000000000000001001000 000
0001010000000001000100101001000000100100100001100001000001000100001100001000100000100001000000000000 11
0100010010100010000000001000010010011001000010000011000010010010010001001000010011001001001000010001 0
0000000000100100000000000110000000001010001000010000000000000001000100000000010110000100110010100000 0001
0001000000100000000010000000100110000001001000011000011001000010000000101000000000000110000000110000 11000
0100000000010110100010000000010010011000001000000001000101001000000000000000110000110010100100100010 010
1000001001000100000010100000001000001000010000000000001001000000100110000000000100010000000001000010 01000
0001011001000010001001010010000000000000001000000010000110000010000110000000110000000010000000000001 000
0000010010000000010000000010011000000000101001100001000100001001001000101001100001100000101001001100 00
0101000000100000000010100100000000000101000100101000101000000010010100000001000000101100000000000000 101000
0000000010010010000010000010000000100101101000000000100100000100010010100000110000010000000000000000 1000
0000110010000000000010000000100100001000000100110100010000100100100000000000010000000100101000100000 10000
0000100001100001101001000000000010010100000001011010000000100100000100000000000100001001000100001100 000
0000000001010011000001010000000100100000000000000000100000100010010000110100000000001000010001000000 0011
0000100000010110000010010000000010010001010001010010001001001000001001010010010001000000000101000000 00
0101000001100001001100001000001101000000000010010000000000000010010100010000100000100110000000000000 01
1001011000000010000011001000010001000000000000100000100100000000100101000000010010100000100010010000 01
0100100000100100000110000000010000000000101100001000000000100010100000001011001000000000001001000000 10010
0100001001100001101100000000000110100100100000100100000001000010000010010001010000000000000000000100 0000
1000001001010011000001001000001000000001100001000100000000000110000010010100100000000011000001001010 000
0100010010000010100100000100110000010000010100000001001000000010011000010001000000000100100000000000 000
0001100000000000000110000000000010011000010000011001010001000000000001001100001000101001000101000001 000
0000010000010000000001010010010001011000000001000001100001000100001101001000000100000100000000000010001
0000000000000110100000000001011010000000100001000001100000100000001100000000000001000100000000000100010
1001001000000101000100000100100101000000000101000000001100000100101001000000010000000010000100010001 00001
0000000000000010010000100100110010000000100110000010010000000000000110000100010000000100100010001000
0100000000000100000000010000110000110000010100100000000010110000010000000100001100001101000000000010
0000001000000000000101000100101000001001000100000100000100101000100000000011000000010000001000001001
0000010000010000001001010000000000101000000100110010010000000100010000000000110000000000000010001001010
0000000010100000100000000000010000000001010001000011000000011000001000001000000000000000001101001000000
1001001000100101100100000000000010100000000010000100100100000010100100110000010010000110000100100000 0
0001010000000000001011001000000000011000000000001010011010010000000000001001000101100000000000000000010
0100010000000100000110000110000100100010100010000010000000000100100000010100000000000001011000001000 0
1000101000100000101000000100100000000100100010100100100010010000100000110010010000000100010010010000 0
0 1 2 3 4 5 6 7 8 9 1

Table 2. The oscillatingly increasing characteristic of the Goldbach-Xu's numbers

$L_r(n)$																														n
1	1	1	1	2	2	2	2	3	2	2	3	2	3	4	1	3	4	3	3	5	4	3	5	3	3	6	2	5	6	1~30
2	5	6	4	5	7	4	4	8	4	4	9	4	4	7	3	6	8	5	5	8	6	7	10	6	5	12	3	5	10	31~60
3	7	9	5	5	8	7	7	11	5	5	12	4	8	11	4	8	10	5	5	13	9	6	11	7	6	14	6	8	13	61~90
5	8	11	6	9	13	8	8	14	6	7	19	6	7	13	6	9	11	7	6	12	9	7	15	9	9	18	8	9	16	90~120
6	9	16	8	8	14	10	8	16	8	9	19	7	10	16	6	14	16	8	12	17	10	8	19	8	10	21	8	10	15	121~150
8	12	17	8	10	15	11	11	20	6	10	24	6	10	19	9	13	17	10	8	16	13	10	20	9	9	22	7	14	18	151~180
8	14	18	10	11	22	13	9	19	11	9	27	11	10	21	6	14	17	11	13	20	13	11	21	10	11	30	10	12	21	181~210
9	14	19	13	11	21	14	13	21	11	13	27	12	11	24	8	16	28	12	12	24	15	13	23	14	10	29	10	14	23	211~240
9	19	22	13	13	23	13	14	27	15	14	32	11	13	23	11	17	24	11	14	25	14	17	22	13	13	30	9	13	30	241~270
11	19	23	10	11	23	18	13	24	12	13	31	11	16	26	12	19	25	12	12	29	16	15	27	12	15	32	12	14	27	271~300
13	20	26	14	19	26	18	16	31	11	16	41	10	13	28	15	18	25	17	16	27	21	15	29	13	19	41	13	16	31	301~330
11	21	33	14	17	28	21	16	30	16	16	39	11	18	30	13	24	31	18	18	24	16	17	37	14	14	39	14	15	31	331~360
15	21	31	14	19	29	18	19	31	17	19	39	14	17	35	14	21	30	17	16	31	26	18	32	16	14	44	14	18	30	361~390
15	22	34	16	14	38	21	15	32	16	14	39	18	20	34	16	20	29	16	21	34	22	22	33	18	16	51	17	17	32	391~420
15	25	31	20	19	39	18	16	33	16	21	46	18	19	36	13	25	39	21	17	37	23	19	34	20	18	48	15	17	34	421~450
15	31	31	19	18	35	23	19	47	17	18	43	17	19	36	18	24	34	18	20	33	25	23	37	19	21	45	16	18	45	451~480
17	27	32	16	19	35	26	16	39	20	23	52	13	25	37	16	28	36	18	17	42	25	23	39	18	19	51	18	22	42	481~510
18	25	36	21	27	40	26	21	39	18	19	57	18	24	44	19	27	37	24	24	39	25	21	40	20	27	54	19	21	39	511~540
18	26	48	22	18	40	28	24	44	25	25	54	16	22	41	22	34	47	19	22	39	26	22	49	23	19	58	17	24	38	541~570
26	27	36	18	22	42	29	25	43	24	22	58	18	21	49	19	26	40	20	19	43	33	23	45	24	23	54	18	28	43	571~600
20	32	42	21	21	49	27	25	45	21	22	55	28	24	42	17	34	44	23	26	45	28	23	51	20	21	68	21	26	42	601~630
21	27	40	26	25	42	27	25	46	21	29	60	23	25	49	20	33	53	24	23	46	30	23	46	27	25	66	20	26	53	631~660
22	41	47	25	25	45	27	27	54	24	22	60	22	20	41	24	33	44	27	26	48	28	27	47	23	27	61	19	24	59	661~690
20	30	44	24	24	45	34	26	48	23	25	58	18	30	47	19	34	41	22	23	57	35	25	50	22	22	60	27	28	45	691~720
20	36	49	25	34	48	33	29	47	24	24	73	22	30	51	25	34	52	29	24	52	31	27	50	28	32	67	26	27	51	721~750
25	33	59	22	26	56	31	26	48	28	27	69	22	32	47	25	46	46	21	29	51	31	28	62	24	25	72	29	28	51	751~780
25	38	57	22	26	47	31	34	58	28	26	71	25	26	64	24	36	53	23	30	47	42	35	53	27	31	65	23	35	55	781~810
29	40	60	27	27	67	35	26	52	25	28	76	28	30	55	26	39	54	28	34	53	37	35	56	28	30	83	24	31	56	811~840
26	37	57	32	28	52	34	30	55	29	32	78	26	26	68	20	38	64	29	30	53	36	25	55	35	33	76	25	33	55	841~870
25	48	52	26	30	55	41	28	69	31	31	73	28	26	53	33	37	59	25	30	52	36	36	66	31	27	75	31	31	72	871~900
28	36	53	27	27	53	46	26	58	33	29	76	25	34	62	27	37	54	32	28	70	38	28	54	32	28	76	34	32	53	901~930
28	43	58	29	36	61	38	32	61	30	28	91	35	31	63	33	36	55	36	29	58	36	30	66	28	35	81	30	34	58	931~960
30	39	68	27	33	65	41	29	58	34	31	83	29	34	53	27	48	56	26	28	62	38	32	71	33	32	82	29	32	58	961~990
30	42	59	26	28	56	37	44	59	35	28	84	27	35	73	27	41	59	32	31	62	49	29	64	33	32	85	29	38	75	991~1020
25	42	65	34	32	75	45	32	58	33	35	82	36	33	58	26	48	59	30	39	62	46	34	62	32	31	97	30	35	71	1021~1050
32	41	67	34	37	63	37	32	67	30	42	82	33	35	66	28	47	81	32	36	63	43	32	64	43	37	79	34	38	67	1051~1080
31	52	65	30	35	68	41	33	85	33	31	82	33	33	62	34	46	60	32	34	60	47	38	68	34	28	90	35	33	78	1081~1110
28	41	67	31	36	58	51	35	77	32	33	86	26	44	70	31	44	75	31	38	75	41	33	61	32	36	92	38	33	68	1111~1140
37	47	68	32	40	63	49	31	68	33	34	114	35	40	66	38	48	67	38	34	64	43	35	68	35	38	94	33	36	76	1141~1170
31	45	83	41	37	62	45	32	66	44	37	92	33	37	76	32	60	67	35	37	67	44	41	84	37	37	90	36	36	72	1171~1200
39	50	69	37	35	78	48	41	72	31	35	94	36	37	86	31	50	75	37	39	73	56	36	71	30	38	93	34	52	69	1201~1230
32	57	71	38	37	87	46	39	76	38	40	94	42	41	75	32	47	68	32	46	74	45	34	72	38	36	112	35	41	68	1231~1260
34	55	70	41	36	70	43	38	77	36	48	99	43	40	71	30	48	82	34	36	64	44	39	86	43	35	95	35	43	73	1261~1290
33	60	68	34	45	71	49	35	91	34	36	98	39	36	72	45	48	77	38	35	69	46	45	76	34	33	107	28	40	81	1291~1320
31	54	80	34	41	70	58	44	72	37	42	97	28	48	77	36	46	74	42	41	88	48	38	75	38	42	100	44	43	87	1321~1350
37	52	73	39	52	78	51	41	79	39	47	128	35	41	82	37	51	79	50	41	75	56	38	86	43	46	109	37	37	72	1351~1380
36	55	101	40	39	68	55	43	81	44	44	104	37	45	78	38	64	72	34	44	90	51	45	97	40	38	104	43	41	76	1381~1410
41	52	78	40	42	89	49	49	78	40	40	110	40	41	100	37	68	78	42	40	71	64	41	76	42	39	107	37	50	90	1411~1440
40	60	74	44	42	94	51	46	91	40	44	99	48	44	83	38	57	80	41	60	80	52	42	82	31	45	122	36	41	87	1441~1470
41	59	75	44	42	82	52	37	98	34	52	116	36	48	86	33	59	95	41	43	79	58	49	84	45	45	104	40	41	78	1471~1500
36	69	76	44	50	85	49	41	99	43	41	110	36	48	92	43	62	93	43	43	83	50	53	78	39	47	113	37	44	102	1501~1530
41	58	79	41	44	83	68	47	83	36	41	109	35	59	84	47	61	92	42	45	100	48	46	82	45	40	117	44	46	85	1531~1560
47	57	83	36	49	80	47	43	75	46	44	138	39	45	85	41	56	85	45	43	93	52	47	83	42	52	112	40	42	87	1561~1590
40	66	98	42	51	90	54	39	83	45	48	110	44	41	87	36	73	80	46	42	86	58	48	112	40	45	112	37	39	82	1591~1620
47	59	87	43	51	76	55	52	89	44	46	110	38	46	112	44	59	86	43	47	78	67	45	90	39	45	120	41	56	93	1621~1650
44	56	92	43	48	107	57	49	84	47	46	116	54	43	89	41	57	84	47	52	89	57	45	94	41	55	138	43	47	105	1651~1680

Note: 1.The max number of $L_r(n)$ is three digits in the table.

2.The nth even number $D(n)=2(n+3)$.

Appendix A

```
      PROGRAM PRIMECAL
      INTEGER W, R, ODD, PRIME, NODD(600000)
      CHARACTER*15 FNAME1
C     THIS IS TO CALCULATE PRIMES UP TO NUPTO

      PRINT *, “INPUT FILE NAME : FNAME1 = D0101T1.DAT=?”
      READ(*,'(A)') FNAME1

      PRINT *, “INPUT NUPTO=?”
      READ *, NUPTO

      DO 1 I1=1, NUPTO
        NODD(I1)=0
1     CONTINUE

      NPRIMES=0
      DO 50 ODD=3, NUPTO, 2
        W=0
        I=2
        J=SQRT(REAL (ODD))
10      R=MOD(ODD, I)
        IF (R .EQ. 0) THEN
          W=1
        ELSE
          I=I+1
        END IF
        IF (I.GT.J .OR. W.NE.0) THEN
        ELSE
          GOTO 10
        END IF
        IF (W .EQ. 0) THEN
          PRIME=ODD
          PRINT *, PRIME
          NODD(PRIME)=1
          NPRIMES=NPRIMES+1
        ELSE
        END IF
50    CONTINUE

      WRITE (*,100) (NODD(ODD), ODD=1, NUPTO, 2)
      PRINT *, NPRIMES

      OPEN(6,FILE=FNAME1,ACCESS='SEQUENTIAL',STATUS='NEW')
      WRITE (6,101) (NODD(ODD), ODD=1, NUPTO, 2)
      CLOSE(6)
100   FORMAT(1X,100I1)
101   FORMAT(100I1)
      END
```

Appendix B

```
      PROGRAM GOXUNUMB
      INTEGER JTABLE1(6000),LT(6000),LHD(6000),LID(6000),DN,DN0,BN,
     +J0(6000),CN0,CN01,W,W1,LR(6000),J(6000),JLT(6000),JTABLE2(6000)
      CHARACTER*15 FNAME1,FNAME2

      WRITE(*,'(1X,A$)') “ INPUT SEQUENCE NUMBER:(n<=5000) =? ”
      READ(*,*) N

C     NEXT IS TO CALCULATE THE LT(n)
      DO 10, I=1,N,1
        II=MOD(I,2)
        IF (II.EQ.0) THEN
          LT(I)=I/2
        ELSE
          LT(I)=(I+1)/2
        ENDIF
10    CONTINUE

C     NEXI IS TO CALCULATE THE LHD(n)
      LHD(1)=0
      LHD(2)=0

      WRITE(*,'(1X,A$)') “ INPUT INPUT-DATA FILENAME = D0101T1.DAT=? ”
      READ(*,'(A)') FNAME1
      OPEN(6,FILE=FNAME1,ACCESS='SEQUENTIAL',STATUS='NEW')
      READ(6,200) (JTABLE2(I),I=1,N,1)
      CLOSE(6)
      WRITE(*,250) (JTABLE2(I),I=1,N,1)

C     %DELLETE OUT FIRST NUMBER 0 IN JTABLE2(I)
      DO 12 I=1,N-1,1
       JTABLE1(I)=JTABLE2(I+1)
12    CONTINUE

      WRITE(*,250) (JTABLE1(I),I=1,N-1,1)
      PAUSE

      LHD(1)=0
      LHD(2)=0

      DO 20, I=3,N-1,1
        II=MOD(I,2)
        IF (II.EQ.0) THEN
C     WHEN n IS AN EVEN NUMBER!
          CN0=LT(I-1)+1
          CN01=JTABLE1(CN0)
          W=2*LT(I)+1
          DN=JTABLE1(W)
```

```
            IF (CNO1.EQ.0) THEN
              IF (DN.EQ.0) THEN
                LHD(I)=LHD(I-1)
              ELSE
                LHD(I)=LHD(I-1)-1
              ENDIF
            ELSE
              IF (DN.EQ.0) THEN
                LHD(I)=LHD(I-1)+1
              ELSE
                LHD(I)=LHD(I-1)
              ENDIF
            ENDIF
          ELSE
C       WHEN n IS AN ODD NUMBER!
            DNO=2*LT(I)
            DN=JTABLE1(DNO)
            IF (DN.EQ.0) THEN
              LHD(I)=LHD(I-1)+1
            ELSE
              LHD(I)=LHD(I-1)
            ENDIF
          ENDIF
20      CONTINUE

        WRITE(*,300) (LHD(I),I=1,N-1,1)
        WRITE(*,'(1X,A$)') “ABOVE DATA IS LHD(N)”
        PAUSE

C       NEXT IS TO CALCULATE THE LID(n)!
        DO 25, I=1,6,1
            LID(I)=0
25      CONTINUE

        DO 30, I=7,N,1
          II=MOD(I,2)
          W1=LT(I)
          BN=JTABLE1(W1)
          IF (II.EQ.0) THEN
            LID(I)=LID(I-1)
          ELSE
            IF (BN.EQ.0) THEN
              LID(I)=LID(I-1)+1
            ELSE
              LID(I)=LID(I-1)
            ENDIF
          ENDIF
30      CONTINUE

        WRITE(*,300) (LID(I),I=1,N-1,1)
        WRITE(*,'(1X,A$)') “ABOVE DATA LID(N)”
```

```
      PAUSE

C     NEXT IS TO CALCULATE THE J(n)~~JLT(I)!
      DO 35, I=1,8,1
        JLT(I)=0
35    CONTINUE

      DO 40, I=9,N-1,1
        II=MOD(I,2)
        L=LT(I)-1

        DO 45, I1=1,3,1
          J(I1)=0
45      CONTINUE

        IF (II.EQ.0) THEN
C     WHEN n IS AN EVEN NUMBER!
          J0(I)=0
          DO 80, I1=4,L,1
            JJ=2*LT(I)+1-I1+1
            JL1=JTABLE1(I1)
            JL2=JTABLE1(JJ)
            IF (JL1.EQ.0 .AND. JL2.EQ.0) THEN
              J(I1)=1
            ELSE
              J(I1)=0
            ENDIF
            J0(I)=J0(I)+J(I1)
80        CONTINUE
          LLT1=LT(I-1)
          LLT2=LT(I-1)+2
          LLJ1=JTABLE1(LLT1)
          LLJ2=JTABLE1(LLT2)
          IF (LLJ1.EQ.0 .AND. LLJ2.EQ.0) THEN
            JLTA=1
          ELSE
            JLTA=0
          ENDIF
          JLT(I)=J0(I)+JLTA
        ELSE
C     WHEN n IS AN ODD NUMBER!
          DO 90, I1=4,L,1
            JJ=2*LT(I)-I1+1
            JL1=JTABLE1(I1)
            JL2=JTABLE1(JJ)
            IF (JL1.EQ.0 .AND. JL2.EQ.0) THEN
              J(I1)=1
            ELSE
              J(I1)=0
            END IF
            J0(I)=J0(I)+J(I1)
```

```
90          CONTINUE
            LLT1=LT(I-1)+1
            LLT2=LT(I-1)+2
            LLJ1=JTABLE1(LLT1)
            LLJ2=JTABLE1(LLT2)
            IF (LLJ1.EQ.0 .AND. LLJ2.EQ.0) THEN
              JLTA=1
            ELSE
              JLTA=0
            END IF
            JLT(I)=J0(I)+JLTA
          END IF
40      CONTINUE

        WRITE(*,'(1X,A$)') “J(N)~~JLT(N)”
        WRITE(*,300) (JLT(I),I=1,N-1,1)
        PAUSE

        DO 60 I=1,N-1,1
          LR(I)=LT(I)-LHD(I)-LID(I)+JLT(I)
60      CONTINUE

        WRITE(*,*) (LR(I),I=1,N-1,1)
        WRITE(*,'(A$)') “ABOVE DATA ARE LR(N)”
        PAUSE
        WRITE(*,'(1X,A$)') “INPUT OUTPUT-DATA FILENAME=? GOLDXUN1.DAT=”
        READ(*,'(A)') FNAME2
        OPEN(8,FILE=FNAME2,ACCESS='SEQUENTIAL',STATUS='NEW')
        WRITE(8,300) (LR(I), I=1,N-1,1)
        CLOSE(8)

200     FORMAT(100I1)
300     FORMAT(2X,30I3)
250     FORMAT(1X,100I1)
        END
```

Reference

[1] Luo-Geng Hua, a lecture in Cambridge University, English, 1979 (unpublished).

[2] Ji-Sheng Na, “An estimation on a sum formula”, KEXUETONGBAO, 9(1986) 641-647.

[3] Wan-Dong Xu, “A new two-dimension sieve method and the proof of the Goldbach's conjecture” (www.paper.edu.cn/0606266) (unpublished).

[4] Wan-Dong Xu, “The irregular distribution of primes up to 300,000 in the sequence of odd numbers” (www.paper.edu.cn/0607179) (unpublished).

Note: we will present two programs in C language for these calculatings in next paper.