

00063D

Program Description I

Program Title **FACTORS AND PRIMES**

Contributor's Name **HEWLETT-PACKARD COMPANY**
Corvallis Division
 Address **1000 N.E. Circle Boulevard**
 City **Corvallis, OR 97330**

State

Zip Code

Program Description, Equations, Variables

This program will find all prime factors of a positive integer n , or list all prime numbers between lower and upper bounds specified by the user.

A routine under LBL a is used in determining the factors of an integer n . This routine selects a trial divisor d and tests d as a factor of n . If d divides n , then $n \leftarrow n/d$ and d is tested as a factor of the new n . If d does not divide n , a new d is selected. The process continues until $d > \sqrt{n}$, at which point n is returned as the final factor. The trial divisor d takes on the values 2, 3, 5, 7; then for $d > 10$, d takes on those values that satisfy $(d - 10) \bmod 30 = 1, 3, 7, 9, 13, 19, 21, \text{ or } 27$. Thus in the first cycle of 30 integers from 11 to 40, d takes on the values 11, 13, 17, 19, 23, 29, 31, and 37. This technique eliminates from the test those values of d ($d > 10$) which are divisible by 2, 3, or 5.

To list primes, a lower bound for the search must be specified. The upper bound is an optional input; if omitted, a default value of 2×10^9 is assumed. Upper and lower bounds need not be integers. The search for primes also uses LBL a to determine if an integer n has any factors or is indeed prime. Once an integer n ($n \geq 3$) has been tested and found to be either prime or non-prime, the next integer tested is $n+2$.

Remarks:

1. The number n to be factored must be an integer in the range $0 < n \leq 2 \times 10^9$. Any other input will result in a program halt with a display of "Error".
2. The upper bound of the search for primes must be greater than or equal to the lower bound, or an Error halt will occur.
3. AUTO mode is available to allow automatic output of all calculated results through PRINT/PAUSE commands. The end of all calculations is signalled by a 0.00 in the display for both modes.
4. Either routine can be quite time-consuming for large integers.

Operating Limits :

This program has been verified only with respect to the numerical example given in *Program Description II*. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

Sketch(es)

Sample Problem(s)

Example 1:

Find the prime factors of 924. Do not set AUTO mode.

Keystrokes:

924	A	→	2.00	
R/S		→	2.00	
R/S		→	3.00	
R/S		→	7.00	
R/S		→	11.00	
R/S		→	0.00	(end)

Thus $924 = 2 \times 2 \times 3 \times 7 \times 11$.**Example 2:**

Find the prime factors of 3623. Do not use AUTO mode.

Keystrokes:

3623	A	→	3623.00	
R/S		→	0.00	(end)

3623 is prime.

Solution(s)

Example 3:

Find all prime numbers between 101 and 125. Use AUTO mode.

Keystrokes:

101	B	125	C	E	→	1.00	(AUTO set)
D					→	101.00	***
						103.00	***
						107.00	***
						109.00	***
						113.00	***
						0.00	(end)

Reference(s)

Program Listing I

Factors and Primes

STEP	KEY ENTRY	COMMENTS	STEP	KEY ENTRY	COMMENTS
001	*LBLA	Factor integer n.	057	RCLB	
002	STOB		058	-	
003	ENT+		059	X<0?	If upper < lower, halt on Error.
004	INT		060	GTOS	
005	X≠Y?	If non-integer, halt on Error.	061	RCLA	
006	GTOS		062	INT	
007	0		063	2	Handle 2 as special case.
008	STOD	Initialize d.	064	X≠Y	
009	X≠Y		065	X=Y?	
010	X≠Y?	If n < 0, halt on Error.	066	GTOD	
011	GTOS		067	2	
012	2		068	+	
013	EEX		069	.	This routine finds greatest potential prime < user's input.
014	9		070	5	
015	X≠Y		071	+	
016	X>Y?	If n > 2 x 10 ⁹ , halt on Error	072	INT	
017	GTOS	F1 clear for factors.	073	2	
018	CF1	Find factors.	074	x	
019	GSB _a		075	1	
020	RTN		076	-	
021	*LBLB	Lower bound for primes.	077	*LBLB	
022	STOA		078	STOE	Store final prime U.
023	X<0?	If negative, halt on Error.	079	RCLA	
024	GTOS		080	RTN	
025	ENT+	This routine finds smallest potential prime > user's input.	081	*LBLD	Routine to list primes.
026	INT		082	0	
027	X=Y?		083	STOD	Initialize d=0
028	GTOD		084	RCLB	
029	1		085	2	If L = 2, print 2, add 1 and go.
030	+		086	X≠Y	
031	*LBLB		087	X=Y?	
032	2	Handle 2 as special case (only even prime).	088	GTOD	
033	X≠Y		089	1	If L = 1, print 1, 2, add 1 and go.
034	X=Y?		090	X≠Y?	
035	GTOD		091	GTOD	
036	2		092	GSB _e	If L ≠ 1 and L ≠ 2, go directly to LBL 1.
037	+		093	+	
038	INT		094	RCLC	
039	2		095	X≠Y	
040	x		096	X>Y?	
041	1		097	GTOD	
042	+		098	*LBLB	
043	*LBLB		099	GSB _e	Output 2.
044	STOB	Store beginning prime L.	100	1	
045	STOC		101	GTOD	
046	2	2 x 10 ⁹ is default upper bound.	102	*LBL1	
047	EEX		103	GSB _a	Begin main loop.
048	9		104	RCLC	Check for factors of current n (R _a).
049	STOE	If input > 2 x 10 ⁹ , halt on Error.	105	RCLC	If R _a = R _c , n is prime.
050	X≠Y?	Flag 1 set for primes.	106	X=Y?	Output n.
051	GTOS		107	GSB _e	
052	SF1		108	2	
053	RCLA		109	*LBLB	
054	RTN		110	+	
055	*LBLC	Upper bound for primes.	111	STOC	Set n to next potential prime.
056	STOA		112	STOB	

REGISTERS									
0	1	2	3	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
A	B	C	D	E	F	G	H	I	J
Used	n	Potential prime	d	U					

Program Listing II

STEP KEY ENTRY COMMENTS STEP KEY ENTRY COMMENTS

113	RCLC		119	RTN	Loop again for next 30.
114	XZY		120	CTD2	-----
115	XZY		121	*LBL3	
116	GT04	If n > U, exit.	122	RCLD	Tests if d/n.
117	0		123	+	
118	STOD	Else loop again.	124	STOD	d ← d + x
119	CTD1		125	RCLB	n
120	*LBL4	Subroutine called from both	126	XZY	
121	2	A & D which finds factors	127	+	n/d
122	CSB3	of n.	128	LSTX	d, n/d
123	X=0?		129	XZY?	If d > n/d, then d > √n
124	RTN		130	CTD0	Exit.
125	1		131	XZY	
126	CSB3	Check first if n divisible by	132	INT	[n/d] := INT (n/d)
127	X=0?	2, 3, 5, or 7.	133	LSTX	n/d, [n/d]
128	RTN		134	XZY?	If non-integer, d does
129	2		135	RTN	not divide n.
130	CSB3	LBL 2 check for division by	136	STOE	Else n ← n/d
131	X=0?	integers whose position in a	137	F1?	
132	RTN	cycle of 30 corresponds to	138	CLX	If finding primes, exit.
133	2	11, 13, 17, 19, 23, 29, 31,	139	F1?	
134	CSB3	or 37.	140	RTN	
135	X=0?		141	RCLD	If factoring, output d.
136	RTN		142	CSBe	
137	*LBL2	-----	143	0	Check for d a multiple
138	4		144	CTD3	factor.
139	CSB3	11 (=7 +4)	145	*LBL0	-----
140	X=0?		146	F1?	Coming here means n
141	RTN		147	CLX	prime.
142	2		148	F1?	If finding primes, exit.
143	CSB3	13 (=11 +2)	149	RTN	
144	X=0?		150	RCLB	Else output last n.
145	RTN		151	CSBe	-----
146	4		152	*LBL4	
147	CSB3	17 (=13 +4)	153	CLX	Exit displaying 0.
148	X=0?		154	F0?	
149	RTN		155	SPC	
150	2		156	RTN	
151	CSB3	19 (=17 +2)	157	*LBL5	
152	X=0?		158	F0?	Auto toggle.
153	RTN		159	CTD0	
154	4		160	SF0	
155	CSB3	23 (=19 +4)	161	1	
156	X=0?		162	RTN	
157	RTN		163	*LBL0	
158	6		164	CF0	
159	CSB3	29 (=23 +6)	165	0	
160	X=0?		166	RTN	
161	RTN		167	*LBL6	
162	2		168	F0?	Output routine.
163	CSB3	31 (=29 +2)		PRTX	Print if AUTO.
164	X=0?			F0?	
165	RTN			RTN	
166	6			R/S	
167	CSB3	37 (=31 +6)		RTN	Halt if not.
168	X=0?				

LABELS					FLAGS		SET STATUS		
A n → Factors	B Primes from	C Primes to	D → Primes	E AUTO?	0 Auto	1 Primes	FLAGS	TRIG	DISP
a Factor n	b	c	d	e Output			0 ☐ ON ☐ OFF	DEG ☐	FIX ☐
0 Used	1 Primes loop	2 Divisor loop	3 d/n?	4 Exit			1 ☐	GRAD ☐	SCI ☐
5 Non-existent	6	7	8 L = 1 or 2	9			2 ☐	RAD ☐	ENG ☐
							3 ☐		n 2