♦MoHPC♠ *The Museum of HP Calculators*

HP Forum Archive 21

[Return to Index | Top of Index]

Programming exercise

Message #1 Posted by ***Gerson W. Barbosa*** *on 22 Aug 2012, 10:00 p.m.*

What is my licence plate number?

Hint #1: The four-digit number is the smallest number expressible as the sum of two cubes in two different ways, that is, n = a^3 + b^3 = c^3 + d^3; a, b, c and d are positive integers;

Hint #2: n < 2000;

Hint #3: the three-letter pefix is CAB backwards :-)

My straightforward approach (not to call it 'brute-force' :-) could be used on the HP-41CV, but I have preferred to do it on my HP 50g. My RPL program is somewhat clumsy because I wrote a QBASIC program first and then converted it to RPL, but it does find n, a, b, c and d in 70 seconds on the real calculator when the input is 1999. When the input is 9999, another larger solution is found, but it takes about 10 minutes before the solutions are shown. The ten solutions less than 100,000 are found in about 9 minutes on the emulator. There is much room for improvement as no mathematical approaches have been tried-- neither could I. (I would have chosen API-3142, but it wasn't available anymore. On second thought I should have chosen something as simple as ABC-1234 :-)

Have fun!

Re: Programming exercise

Message #2 Posted by ***Don Shepherd*** *on 22 Aug 2012, 10:45 p.m.,*
in response to message #1 by Gerson W. Barbosa

So how long have you been driving a taxicab in a suburb of London?

Re: Programming exercise

Message #3 Posted by ***Gerson W. Barbosa*** *on 22 Aug 2012, 11:15 p.m.,* *in response to message #2 by Don Shepherd*

Ah, you know the story! Well, most everyone here does, I think :-)

Re: Programming exercise

Message #4 Posted by ***Valentin Albillo*** *on 23 Aug 2012, 4:27 a.m.,*
in response to message #3 by Gerson W. Barbosa

> Quote:
>
> Ah, you know the story! Well, most everyone here does, I think :-)

Ramanujan, perchance ?

Best regards from V.

Re: Programming exercise

Message #5 Posted by ***LHH*** *on 23 Aug 2012, 4:46 a.m.,*
in response to message #4 by Valentin Albillo

This is just the kind of problem I like to try to solve in my head when I go to bed. Usually it's calculating the time to a star at various speeds and distances (or something along those lines). I may have mentioned before my (young) kids and I once tried to estimate the number of atoms in the universe and came up with 10^75. Recently I saw an estimate of 10^80 elemental particles so we actually came pretty close! This would have been more difficult just remembering all the possible cubes but I'm sure it would have put me to sleep quickly (which is the desired effect)! Anyway, just doodling I wrote down all the possible cubes and have an answer (BAC-1729) but I did cheat by using a biological computer (I will accept my disqualification in a manly fashion).

Re: Programming exercise

Message #6 Posted by ***Valentin Albillo*** *on 23 Aug 2012, 5:59 a.m.,*
in response to message #5 by LHH

. Hi, LHH: .

> Quote:

> This is just the kind of problem I like to try to solve in my head when I go to bed.

Hehe, just in case you're too sleepy to give it a try, this short HP-71B code will find the numeric part of the answer in negligible time:

```
>CAT
workfile   BASIC   150 07/26/04 15:00


>LIST


10 DESTROY ALL @ INPUT "Max=";M @ R=INT(M^(1/3)) @ DIM S(2*R*R*R)
20 FOR I=1 TO R @ FOR J=I TO R @ N=I*I*I+J*J*J
30 IF S(N) THEN DISP N;"is the minimum" @ END ELSE S(N)=1
40 NEXT J @ NEXT I @ DISP "None found up to";M


>RUN
Max=1000
None found up to 1000


>RUN
Max=2000
 1729 is the minimum
```

It's a trivial brute-force approach and it certainly could be made faster by simply pre-computing a cubes table up to the required limit so that N=I*I*I+J*J*J would be computed instead as N=C(I)+C(J) and other such tricks but it's pretty pointless as the running time is already completely negligible.

Other trivial optimizations would include:

- taking out of the J loop the invariant computation of I*I*I or C(I) if using the cubes table

- using a string S$ of the correct length instead of a vector filling it up initially to all "0" characters, afterwards storing a "1" to mark each achieved sum in the correct position within the string after previously checking if the position is already marked so that a minimum solution has been found. This would save lots of memory as the string would take much less space than the array (which could have been declared INTEGER to save about 50% memory, by the way).

- if your HP model has the equivalent of flag or bit arrays, this would be ideal, as the S array or the S$ string are just mimicking them etc, etc, etc, but as it's so trivial and runs instantly I didn't bother, too lazy today ... XD

Best regards from V.

Edited: 23 Aug 2012, 6:11 a.m.

Re: Programming exercise

Message #7 Posted by ***Gilles Carpentier*** *on 23 Aug 2012, 10:53 a.m.,*
in response to message #6 by Valentin Albillo

HI

```
10 DESTROY ALL @ INPUT "Max=";M @ R=INT(M^(1/3)) @ DIM S(2*R*R*R)
20 FOR I=1 TO R @ FOR J=I TO R @ N=I*I*I+J*J*J
30 IF S(N) THEN DISP N;"is the minimum" @ END ELSE S(N)=1
40 NEXT J @ NEXT I @ DISP "None found up to";M
```

Interesting. But how are sure that N is the minimum ?

or:

20 FOR I=1 TO R-1 @ FOR J=I+1 TO R @ N=I*I*I+J*J*J ?

Re: Programming exercise

Message #8 Posted by ***C.Ret*** *on 23 Aug 2012, 11:34 a.m.,*
in response to message #6 by Valentin Albillo

Hi.

I used the same algorithm as you on my Commodore C128D. In your code, you use the aray S() to "flag" with bi-cubic positions have already bee encounted. That was my first approach idea. But, using a whole integer array to only record '0' and '1' appears to me as a too binary. Instead of the '1' flag, I put the a value (a>0), that makes possible a more explicit display of the results n = a,b = c ,d :

```
LIST
10 A=0:B%=0:N=0:INPUT "Max:";M%:R%=(M%/2)^(1/3)
20 DIM P%(M%),C(2*R%)
30 FOR N=1 TO 2*R%:C(N)=N*N*N:NEXT N
```

```
40 FOR A=1 TO R:B%=A:DO:B%=B%+1:N=C(A)+C(B%):IF N>M% THEN EXIT
50 IF P%(N) THEN PRINT N,"="A;B%"="P%(N);INT((N-P%(N))^(1/3)):ELSE P%(N)=A
60 LOOP:NEXT A:END


READY.
RUN
Max:? 32000
 1729 = 9  10 = 1  12
 4104 = 9  15 = 2  16
 13832 = 18  20 = 2  24
 20663 = 19  24 = 10  27
READY.
```

Note that it takes a few seconds on a Commodore C128/C128D so that pre-computing the cubes (line 30) makes no significant changes. Except perhaps better arithmetic precision, the Commodore 8bits are well known for really bad math! Max limit is 32000 due to conventional memory limit (128 ko). Max limit may be extend using memory add on such as graphic memory or unconventional RAM banks (but also need PEEK/POKE bad programming practice – current practices at C64 times!).

Edited: 23 Aug 2012, 11:40 a.m.

Re: Programming exercise

Message #9 Posted by ***Gerson W. Barbosa*** *on 23 Aug 2012, 12:30 p.m.,*
in response to message #8 by C.Ret

When I see this and other approaches here, mine looks quite silly! It could be done within two loops only once, instead I used two nested loops twice. Anyway, here it is, just for the records:

```
%%HP: T(3)A(R)F(,);
\<< 3, XROOT IP \-> r1
  \<< { } 1, r1
    FOR a a r1
      FOR b a SQ a * b SQ b * + +
      NEXT
    NEXT OBJ\-> \->ARRY r1 SQ r1 + 2, / 1, r1 1, - \-> n e1 s1
    \<< 1, 1, n 1, -
      FOR i GETI UNROT i r1 1, + ==
        IF
        THEN s1 'r1' STO+ 1, 'e1' STO+ 's1' 1, STO-
        END e1 r1 s1 \-> e2 r2 s2
        \<< OVER i 1, + DUP n
```

```
          FOR j GETI j r2 1, + ==
            IF
            THEN s2 'r2' STO+ 1, 'e2' STO+ 's2' 1, STO-
            END 6, PICK ==
            IF
            THEN 5, PICK 1, IQUOT \->STR "=" + e1 1, IQUOT \->STR + "^3+" + 6, PICK e1 3, ^ - 3, XROOT 1, IQUOT \->STR
              + "^3=" + e2 1,IQUOT \->STR + "^3+" + 6, PICK e2 3, ^ - 3, XROOT 1, IQUOT \->STR + "^3" + 6, ROLLD
            END
          NEXT
        \>> DROP2 ROT DROP
      NEXT DROP
    \>>
  \>> DROP
\>>


40000 TN -->  7: "1729=1^3+12^3=9^3+10^3"
              6: "4104=2^3+16^3=9^3+15^3"
              5: "13832=2^3+24^3=18^3+20^3"
              4: "39312=2^3+34^3=15^3+33^3"
              3: "32832=4^3+32^3=18^3+30^3"
              2: "40033=9^3+34^3=16^3+33^3"
              1: "20683=10^3+27^3=19^3+24^3"


(after whopping 154 seconds on the emulator!)
```

Re: Programming exercise

*Message #10 Posted by **Gerson W. Barbosa** on 23 Aug 2012, 12:44 p.m.,*
in response to message #5 by LHH

> Quote:
>
> Anyway, just doodling I wrote down all the possible cubes and have an answer (BAC-1729) but I did cheat by using a biological computer

Your superior carbon-based computer is absolutely right! I ought to have tried using mine as well :-)

Re: Programming exercise

*Message #11 Posted by **Gerson W. Barbosa** on 23 Aug 2012, 12:49 p.m.,*
in response to message #4 by Valentin Albillo