HP Forum Archive 13

[Return to Index | Top of Index]

Matrix Reloaded

*Message #1 Posted by **Valentin Albillo** on 26 May 2003, 7:43 a.m.*

Hi to all of you, specially to HP-15C lovers:

Here's a little HP-15C Quiz/Challenge which doesn't just try to be merely enjoyable but useful as well. Though it is proposed and particularized for the venerable HP-15C, the intended solution will be equally applicable to all other HP calculators with matrix capabilities, i.e.: HP42S, HP-71B+Math ROM, and HP-41C+Advantage ROM, among others.

The Quiz/Challenge

Suppose we're dealing with an HP-15C's application that requires a number of matrices holding id codes of some sort, represented as 10-digit integer values from -1E9 to +1E9, say.

The challenge is to write a *general purpose* subroutine which takes two matrix descriptors placed in the X,Y stack registers and must then swap the values of their corresponding matrix elements, subject to these requirements:

1) It must work with *any two distinct arbitrary matrices* placed in X,Y, (i.e: A,B,C,D, or E), of the same, arbitrary dimensions up to a *minimum of 30 elements per matrix*. The routine must not assume any two particular matrices nor any specific dimensions.

2) It must be as short and fast as possible, and use as few registers and resources as possible, preferably just the stack.

For example, if you have C = (3, 1, 4) and D = (2, 7, 1), the code segment:

```
RCL MATRIX C
RCL MATRIX D
GOSUB A
```

(where LBL A, ..., RTN is your subroutine) should result in matrix C and matrix D holding now (2, 7, 1) and (3,1,4), respectively.

Try your hand at it. For the record, my best solution to date is an 11-step routine (not including LBL & RTN), which swaps two 5x5 matrices in 3.7 seconds, and uses no register at all, not even the index registers R0, R1 and RI, just the stack. Let's see yours :-)

Here is my solution

*Message #2 Posted by **Patrick** on 26 May 2003, 2:40 p.m.,*
in response to message #1 by Valentin Albillo

Valentin

Great problem! Anything to keep interest in my beloved HP-15C is very welcome.

Below is my solution. I haven't timed it yet, so I don't know if it is faster or slower than yours, however it is shorter. Not including LBL/RTN it is 8 steps. Like yours, it uses just the stack, no other resources. It even preserves the Z and T registers!

I never would have thought about this solution had you not mentioned how short was the one you had!

Anyhow, here it is for your consideration ...

```
LBL E
GSB .5
GSB .5
LBL .5
STO RESULT
x<>y
CHS
+
LAST x
RTN
```

Some words of explanation... I chose "E" for "Exchange". I also have the convention to use the dot labels for private routines inside programs and to use the non-dot labels for routines I expect to call from the keyboard. I use 5 because it is private to routine E, the 5th lettered subroutine.

I will post a follow-up where I explain the workings of the routine and post timings.

Thanks again, Valentin... I much enjoyed that!

BTW... I do notice the beginnings of the values pi and e in your example... you should have used your 15C's RAN# function!

Re: Here is my solution

*Message #3 Posted by **Patrick** on 26 May 2003, 5:08 p.m.,*
in response to message #2 by Patrick

Well, it turns out that this solution is considerably slower than yours. For two 5x5 matrices, it takes a little over 6 seconds to execute, compared with your 3.7 seconds. Not having seen your solution yet, I won't hypothesize on the reasons.

True numerical analysts in the audience will also notice that this is truly not a correct solution. It depends on the fact that b = (a + b) - a, which is not always true on a calculator if a and b differ wildly in magnitude (e.g., a=1.0e10, b=1.0e-10).

As to the workings of the algorithm, it is clearly seen that the purpose of LBL E is to simply execute the subroutine LBL .5 exactly three times. Each time you execute this subroutine, the values in the matrices (say, X= matrix A and Y = matrix B) get changed as follows:

```
A <-- A-B
B <-- -B
```

Also, at the end of the routine, the stack is changed so that the descriptors are flipped: X = matrix B, Y = matrix A.

If you do this operation exactly three times, keeping in mind this role reversal of A and B each time, you will see that the contents of A and B are interchanged, thus:

```
A  --> A-B --> B-A --> B
B  --> -B  --> -A  --> A
```

QED.

Re: Here is my solution

*Message #4 Posted by **Valentin Albillo** on 28 May 2003, 4:36 a.m.,*
in response to message #3 by Patrick

Hi Patrick,

Thank you for your input to my Quiz/Challenge, I agree with your remark about raising interest in the HP-15C, it's a real pity this *ubercalculator* has become so unaffordable, resulting in fewer HP calc fans finding it possible to get hold of one ...

Your solution is truly excellent, I don't think a shorter one is possible though my solution, while less elegant, is faster:

```
  LBL A
  STO RESULT
  X<>Y
   +
```

```
RCL RESULT
LAST X
STO RESULT
 -
X<>Y
STO RESULT
X<>Y
 -
RTN
```

That's 11 steps (not including LBL/RTN), and swaps two 5x5 matrices in 3.7 seconds in my HP-15C (by the way, yours takes 5.7 seconds in my machine). Why is it faster ? Several reasons:

- my solution has *no branching at all*, it runs straight, thus avoiding the overhead of having to execute two GOSUB, three LBL, and three RTN, as your does. The instruction count reveals that your routine executes *24 instructions* in all, while mine executes only 13. Also, all my instructions are 1 byte each, while your GSB .5 and LBL .5 are *2-bytes each* (the manual's wrong !), so the executed byte count actually comes out as *29 vs. 13*. This accounts for an extra 0.8 seconds. Simply by changing your LBL .5 and GSB .5 to LBL 5 and GSB 5 you would save 3 bytes of program storage (23%), and 5 bytes of execution time.

- my solution performs the swap like this, in HP-71 BASIC code:

  ```
  MAT X = X + Y
  MAT Y = X - Y
  MAT X = X - Y
  ```

  so it's doing two subtractions and one addition, for a total of 75 arithmetic operations in the case of 5x5 matrices. On the other hand yours performs three additions, which amount to 75 arithmetic operations as well, *but also three CHS*, which means changing the sign of said 75 elements. This accounts for the remaining 1.2 additional seconds.

By the way, both your solution and mine are *truly correct* solutions under the quiz conditions, which specified that the matrices would hold *integer* values from -1E9 to +1E9. For those values, B=(A+B)-A holds on an HP-15C.

For arbitrary values, the smaller ones could be rounded or lose accuracy if very different scales are involved. Depending on the application, this could be a *desirable* side effect (!), converting a matrix with very small elements (perhaps some increment in a Newton method applied to matrix equations) in a zero matrix, thus nicely fulfilling the criteria to end the iterations !

However, both your solution and mine have a *weakness* which is easily cured: though they meet the quiz requirements ("two *distinct* matrices"), the fact is that if both matrices X, Y are one and the same, then the matrix will be *zeroed* ! The remedy is to insert these two steps right after the routine's label:

```
TEST 5
RTN
```

this compares both matrix descriptors for equality, and if they happen to be the same, the routine returns immediately, thus leaving the matrix elements *unchanged*, as it should.

A final remark: HP-71/42S/41C owners, take note. The fastest, best way to exchange two matrices is by making use of an auxiliary matrix T, like this:

```
MAT T = X
MAT X = Y
MAT Y = T
```

but if the matrices are so large that there isn't enough RAM to allocate T, the technique above:

```
MAT X = X + Y
MAT Y = X - Y
MAT X = X - Y
```

can save the day if rounding is not an issue. This also works for the HP42S and HP-41C + Advantage ROM. The technique can be also applied to arbitrary objects as long as they allow addition and subtraction.

Best regards.

Re: Here is my solution

*Message #5 Posted by **Patrick** on 28 May 2003, 2:38 p.m.,*
in response to message #4 by Valentin Albillo

Perfect solution, Valentin. The extra few bytes of length over mine are well, well worth it from a pragmatic point of view.

Your comment about the technique applying to other machines and other types of objects which allow arithmetic, keyed a memory for me from my assembler class so many years ago. A standard technique for swapping the values in two CPU registers without using any other memory is to use the "exclusive or" operator three times:

```
XOR A,B
XOR B,A
XOR A,B
```

Assuming XOR does an "exclusive or" of the two named registers and puts the result in the first register, the effect of these three (very quick) machine instructions is to exactly swap the contents of registers A and B, without using a temporary register.

The neat thing about this litte routine is that, to prove it to yourself that it works, you need only consider the simple case of one bit registers. If it works for the four possible combinations in that case, it must work on arbitrary sized registers since the XOR operation is performed bitwise.

Not every having owned a 42S (but still trying!), I don't know if XOR is an operation you can perform on matrices? I doubt it, but if so, this would likely lead to a faster and shorter routine.

Matrices in the HP42S

*Message #6 Posted by **Vieira, Luiz C. (Brazil)** on 28 May 2003, 9:58 p.m.,*
in response to message #5 by Patrick

Matrices in the HP42S

Hi Valentin, Patrick, guys;

Compared to the HP15C, matrices are treated differently in the HP42S. You may have two nameless matrices in both X- and Y-registers, i.e., there's no need for a "descriptor" as it happens in the HP15C. In the HP42S, you may have as many matrices as memory can hold, each with your particular choice as a name, and other four other matrices (five, with LASTx) stored in stack registers. It's about the same that happens in the HP48/49: In both systems, matrices stored in the stack have no name.

Based on these facts, swapping all elements from one matrix to another is useful only in the HP15C, that has a predefined set of five "names". I take both solutions as too clever, I could not even start thinking of how to do it. I didn't thought about using RESULT, I tried multiplying by neutral elements but did not succeed anyway. As I saw I could not provide a thoughtful answer, I decided to watch only. I'd like having a good idea to show, but I did not.

I also thought that a particular need to swap elements from one matrix to another in the HP15C was to preserve RESULT setting, but both solutions change RESULT contents.

My congrats to both of you.

Luiz C. Vieira - Brazil

Losing RESULTs

*Message #7 Posted by **Patrick** on 28 May 2003, 11:17 p.m.,*
in response to message #6 by Vieira, Luiz C. (Brazil)

Both programs leave at least one stack register unmolested. This means that it is possible to preserve RESULT if necessary. One need only RCL RESULT near the beginning of the routine and STO RESULT at the end, sandwiched between whatever Roll-Up or Roll-Down commands are necessary to manipulate the stack correctly.

On the other hand, I don't know if preserving RESULT is worth it. It seems to me that the user should expect RESULT to be lost in any non-trivial matrix subroutine call. It is like the LAST X of matrix operations.

I would be more inclined to document the fact that RESULT is lost, but that (in the case of Valentin's routine, for example) the original Z register is preserved in Y, Z, and T. The user can then choose whether or not to do the RCL RESULT / STO RESULT gymnastics surrounding the call to the matrix exchange subroutine.

A few RESULT comments

*Message #8 Posted by **Vieira, Luiz C. (Brazil)** on 29 May 2003, 1:57 a.m.,*
in response to message #7 by Patrick

Hi Patrick, folks;

I do not intend to focus a small grain over the whole shore, but I think that if you do not use RESULT correctly, some operations may not occur.

You wrote:

It seems to me that the user should expect RESULT to be lost in any non-trivial matrix subroutine call. It is like the LAST X of matrix operations.

Once, when using the HP15C for the first times, I tried a matrix operation and got successive Error 10 messages. I tried hard to find what was going on and did not get it. After a Pr Error forced condition ([ON]/[-]) it got back to normal, meaning that the exact sequence I tried before could find space to be accomplished. Right after that I saw that I was missing something because RESULT was not properly set. I realized that in my previous unsuccessful matrix operation I did not set RESULT too. And that might be the problem: I was using almost all available registers, and remaining memory was not enough to create a new matrix to hold the resulting one. If I had simply set RESULT as one of the matrices I was working on, I'd have no problem at all.

I take much care about REULT in the HP15C because of this: is not set properly you may generate an error, have no space to accomplish the task or need to search for the result in another matrix. I'm mentioning this all because, as you wrote preiously, RESULT settings for me are not exactly LASTx, instead it's a ***floating*** *X-register matrix setting* because it sets where will the result be placed, and it's a different concept of LASTx, I think.

For me, the hP15C is the best stand-alone calculator ever, and these particularities tease me even more, so I have to read again and again and test new key sequences. As if I did not like this...

Best regards.

Luiz C. Vieira - Brazil

Re: A few RESULT comments

*Message #9 Posted by **Patrick** on 29 May 2003, 4:23 p.m.,*
in response to message #8 by Vieira, Luiz C. (Brazil)

Yes, you are right, Luiz, when you say RESULT plays the role for matrices what a floating X register would do for numbers.

My assertion was about the persistence vs. transience of RESULT. What I mean by this is that, when I call a typical numeric subroutine I do not expect that LAST X will be preserved when the subroutine returns. The value stored there is quite transient. It needs to be in order to play the role that is intended for it. Likewise, I believe the RESULT setting should be viewed in the same manner when calling a matrix based subroutine. In order to do anything non-trivial with matrices, the subroutine programmer will very likely have to set RESULT to something of his/her choosing. When the subroutine returns, I should *expect* that the previous value of RESULT is lost.

Regards, Patrick

O.T.: grammar!

*Message #10 Posted by **Vieira, Luiz C. (Brazil)** on 29 May 2003, 12:41 a.m.,*
in response to message #6 by Vieira, Luiz C. (Brazil)

Hi, folks;

completely off-topic, but I do not like reading my own posts and finding errors, mostly spelling and grammar errors. When there is no answer, I usually purge it, correct it and post it again.

This one has a brutal "**I didn't thought**" amongst others, and I could not believe in my eyes.

Sorry!

Luiz C. Vieira - Brazil

Re: O.T.: grammar!

*Message #11 Posted by **Valentin Albillo** on 29 May 2003, 4:19 a.m.,*
in response to message #10 by Vieira, Luiz C. (Brazil)

Hi, Luiz. You posted:

"I could not believe in my eyes."

It should be:

"I could not believe my eyes.", without the "in" :-)

Best regards.

Learning and learning...

*Message #12 Posted by **Vieira, Luiz C. (Brazil)** on 29 May 2003, 6:37 a.m.,*
in response to message #11 by Valentin Albillo

Thank you! :o)

Re: O.T.: grammar!

*Message #13 Posted by **Ernie Malaga** on 29 May 2003, 3:49 p.m.,*
in response to message #10 by Vieira, Luiz C. (Brazil)

Luiz:

Not that you've written it yourself, but I want to mention several very common mistakes:

1. "It's" instead of "its."

2. "I could care less" instead of "I could not [or couldn't] care less."

3. "Alright" instead of "all right."

4. "There" instead of "their."

5. "Too" instead of "to" (and vice-versa).

...and everybody's nemesis...

6. "Who" instead of "whom" (and vice-versa).

Generally it's not the writer's fault, but the language's. English has to be _the_ Western language that is most difficult to learn. Spelling and pronunciation alone are already chaotic. Add to this a grammar full of exceptions (and exceptions of exceptions), and it's not surprising that we all make mistakes when using the language.

Like you, Luiz, I wasn't born in an English-speaking society. My native Peru speaks Spanish, and it's been an uphill battle to learn this #@*$!! language. (Sorry about my "French.")

I'm still trying to find an exception-free to help me determine when it's "I" and when it's "me" in sentences such as "you and me" (or "you and I"). Sometimes it gets so confusing that I wish we all spoke Esperanto. At least then there would be no irregularities to worry about!

-Ernie

Re: O.T.: grammar!

*Message #14 Posted by **Patrick** on 29 May 2003, 4:18 p.m.,*
in response to message #13 by Ernie Malaga

You didn't mention "your" and "you're"... one of my perennial favourites.

I am a native English speaker but not a school teacher. That said, I believe the rule regarding "I" vs. "me" is relatively simple: you should use whatever you would use if there was not another person being mentioned in the sentence.

For example, you wouldn't say "Me went to the store" so you also don't say "Fred and me went to the store."

Even if there are exceptions to this rule (and I actually doubt it), I'm pretty sure no one would complain about this language style in everyday use.

Re: O.T.: grammar!

*Message #15 Posted by **Dave Shaffer** on 29 May 2003, 5:04 p.m.,*
in response to message #13 by Ernie Malaga

Ernie,

re: "I'm still trying to find an exception-free to help me determine when it's "I" and when it's "me" in sentences such as "you and me" (or "you and I"). Sometimes it gets so confusing that I wish we all spoke Esperanto. At least then there would be no irregularities to worry about!"

The absolutely infallible rule is the "I" is subjective (i.e. the subject of a sentence) and "me" is objective (i.e. the object of a sentence or preposition).

Therefore, you use "I" for the subject of a sentence (as decribed by Patrick: Fred and I went to the store - or You and I went to the store.) whether there is one or more than one subject.

Almost everywhere else, you should use "me." As in, "They gave it to me" (again, whether there are one or more than one object - They gave a dinner for Fred and you and me).

The screw up results when what at first glance looks like it is an object should be subjective. This occurs mostly as a pronoun after the verb "to be." Proper: It is I. Improper: It is me. However, this latter form is VERY common in everyday English usage. Therefore, most English teachers try to stamp it out by correcting at every chance possible - from "It is me" to "It is I." I think the net effect is on those who don't think about it carefully, and who are then scared into thinking that they can never use "I"!!!!

For what it is worth, "who" and whom" are in EXACTLY the same category. "Who" is the subjective form and "whom" is the objective form. If you remember that, you can never go wrong.

All that said, your English is way better than my attempts at foreign language (after five years of French and one of German)!

Re: O.T.: grammar!

*Message #16 Posted by **Ernie Malaga** on 30 May 2003, 3:16 a.m.,*
in response to message #15 by Dave Shaffer

>All that said, your English is way better than my attempts at foreign language (after five years of French and one of German)!

Why, thank you. I've lived in the U.S. since 1979 and have always made an effort to _not_ use Spanish -- not easy living in places like Los Angeles and San Diego (CA) and Miami (FL), the 3 places where I've lived the longest.

You want to know another English grammar thorn? How about "which" and "who"? Even professional writers have trouble with this one!

-Ernie

Re: O.T.: grammar!

*Message #17 Posted by **James M. Prange** on 1 June 2003, 4:15 a.m.,*
in response to message #15 by Dave Shaffer

I agree that "I", "he", and "she" are nominative (or subjective, if you prefer) case, and "me", "him" and "her" are objective case. Any grammarian will tell you that this holds true even for compound subjects or objects.

Personally, I use the rather simple rule of using the word that would "sound right" if I were to substitute a non-compound subject or object. At least, I think that I use this rule; it may be that I sometimes get it wrong when I don't have time to analyse which would be correct, as in casual speaking.

On the other hand, in actual English, in compound subjects, "me", "him", or "her" is very commonly used, probably even more commonly than the "grammatically correct" "I", "he", or "she", and this has apparently been the case for centuries.

Of course, English teachers try to drum the rule into our heads, insisting that we say "He and I...." rather than "Me and him....", but it's as if many people can't seem to learn to use the "correct" nominative case in compound subjects without going too far and using it incorrectly in compound objects. The result is that it's not uncommon to hear (or read) "I", "he", or "she" in compound objects, which grates on my ears.

Could it be that there's an unwritten rule in English that the objective case is used in compound subjects, and that the grammarians have simply never acknowledged this rule? Should grammatical rules describe the actual language, as complicated as it is, or should they be relatively simple, straightforward, easy to write rules that make sense to the people who make the effort to write them down for the rest of us?

That said, sometimes an entire clause that contains its own subject is the object of a sentence; similarly, sometimes an entire clause that contains its own object is the subject of a sentence. These situations can be a bit tricky, and often the wrong case is used in spoken English and casual (and sometimes even formal) writing.

Regards,
James

Re: O.T.: grammar!

*Message #18 Posted by **Ellis Easley** on 1 June 2003, 2:06 p.m.,*
in response to message #17 by James M. Prange

Quote:

> ... it's as if many people can't seem to learn to use the "correct" nominative case in compound subjects without going too far and using it incorrectly in compound objects. The result is that it's not uncommon to hear (or read) "I", "he", or "she" in compound objects, which grates on my ears.

I think this happens when people are trying hard to speak correctly and the one rule pops into their heads at the wrong moment!

[Return to Index | Top of Index]

Go back to the main exhibit hall