CPSC 240: Computer Organization and Assembly Language
Assignment 02, Fall Semester 2023

CWID:885857847_____ Name:Kush Patel_____

1. Download the "CPSC-240 Assignment02.docx" document.
2. Design the "addition.asm" program, and use assembly language to realize the function of the following C++ instructions.

```
unsigned short num1 = 50000;
unsigned short num2 = 40000;
unsigned int sum = 0;
sum = int(num1 + num2);
```

3. Assemble the "addition.asm" file and link the "addition.o" file to get the "addition" executable file.
4. Run the "addition" file with the DDD debugger to display the simulation results of num1 and num2, as well as the simulation results of sum.
5. Insert source code (addition.asm) and simulation results (DDD debugger window) of the memory (num1, num2, and sum) in the document. Use calculator or hand calculation to verify the simulation results.
6. Design the "subtraction.asm" program, and use assembly language to realize the function of the following C++ instructions.

```
signed short num1 = 20000;
signed short num2 = 30000;
signed int dif = 0;
dif = int(num1 - num2);
```

7. Assemble the "subtraction.asm" file and link the "subtraction.o" file to get the "subtraction" executable file.
8. Run the "subtraction" file with the DDD debugger to display the simulation results of num1 and num2, as well as the simulation results of diff.
9. Insert source code (subtraction.asm) and simulation results (DDD debugger window) of the memory (num1, num2, and dif) in the document. Use calculator or hand calculation to verify the simulation results.
10. Save the file in pdf format and submit the pdf file to Canvas before 23:59 pm on 09/13/2023.

[Insert addition.asm here]

```
; addition.asm
; unsigned short num1 = 50000;
; unsigned short num2 = 40000;
; unsigned int sum = 0;
; sum = int(num1 + num2);
```

```
section .data
    num1 dw 50000
    num2 dw 40000
    sum  dd 0
section .text
    global _start
_start:
    mov dx, 0
    mov ax, word[num1]          ;al = num1 = 64h
    add ax, word[num2]          ;al = al + num2 = 2Ch
    adc dx, 0               ;ah = ah + 0 + CF = 01h
    mov word[sum+0], ax         ;sum = al = 9ch
    mov word[sum+2], dx         ;sum = ah = 0ffh
                        ;mov word[sum], ax
    Mov rax, 60             ;terminate excuting process
    mov rdi, 0              ;exit status
    syscall
```

[Insert addition simulation result here]

```
 1 ; addition.asm
 2 ; unsigned short num1 = 50000;
 3 ; unsigned short num2 = 40000;
 4 ; unsigned int sum = 0;
 5 ; sum = int(num1 + num2);
 6 section .data
 7         num1 dw 50000
 8         num2 dw 40000
 9         sum  dd 0
10 section .text
11         global _start
12 _start:
13         mov     dx, 0
14         mov     ax, word[num1]                  ;al = num1 = 64h
15         add     ax, word[num2]                  ;al = al + num2 = 2Ch
16         adc     dx, 0                           ;ah = ah + 0 + CF = 01h
17         mov     word[sum+0], ax                 ;sum = al = 9ch
18         mov     word[sum+2], dx                 ;sum = ah = 0ffh
19                                                 ;mov word[sum], ax
20         Mov     rax, 60                         ;terminate excuting process
21         mov     rdi, 0                          ;exit status
22         syscall
23
24
```

```
(gdb) break 20
Breakpoint 1 at 0x401028: file addition.asm, line 20.
(gdb) x/uh &num1
0x402000:       50000
(gdb)
```

0x402000: 50000

```
 1 ; addition.asm
 2 ; unsigned short num1 = 50000;
 3 ; unsigned short num2 = 40000;
 4 ; unsigned int sum = 0;
 5 ; sum = int(num1 + num2);
 6 section .data
 7         num1 dw 50000
 8         num2 dw 40000
 9         sum  dd 0
10 section .text
11         global _start
12 _start:
13         mov     dx, 0
14         mov     ax, word[num1]                  ;al = num1 = 64h
15         add     ax, word[num2]                  ;al = al + num2 = 2Ch
16         adc     dx, 0                           ;ah = ah + 0 + CF = 01h
17         mov     word[sum+0], ax                 ;sum = al = 9ch
18         mov     word[sum+2], dx                 ;sum = ah = 0ffh
19                                                 ;mov word[sum], ax
20         Mov     rax, 60                         ;terminate excuting process
21         mov     rdi, 0                          ;exit status
22         syscall
23
24
```

```
(gdb) x/uh &num1
0x402000:       50000
(gdb) x/uh &num2
0x402002:       40000
(gdb)
```

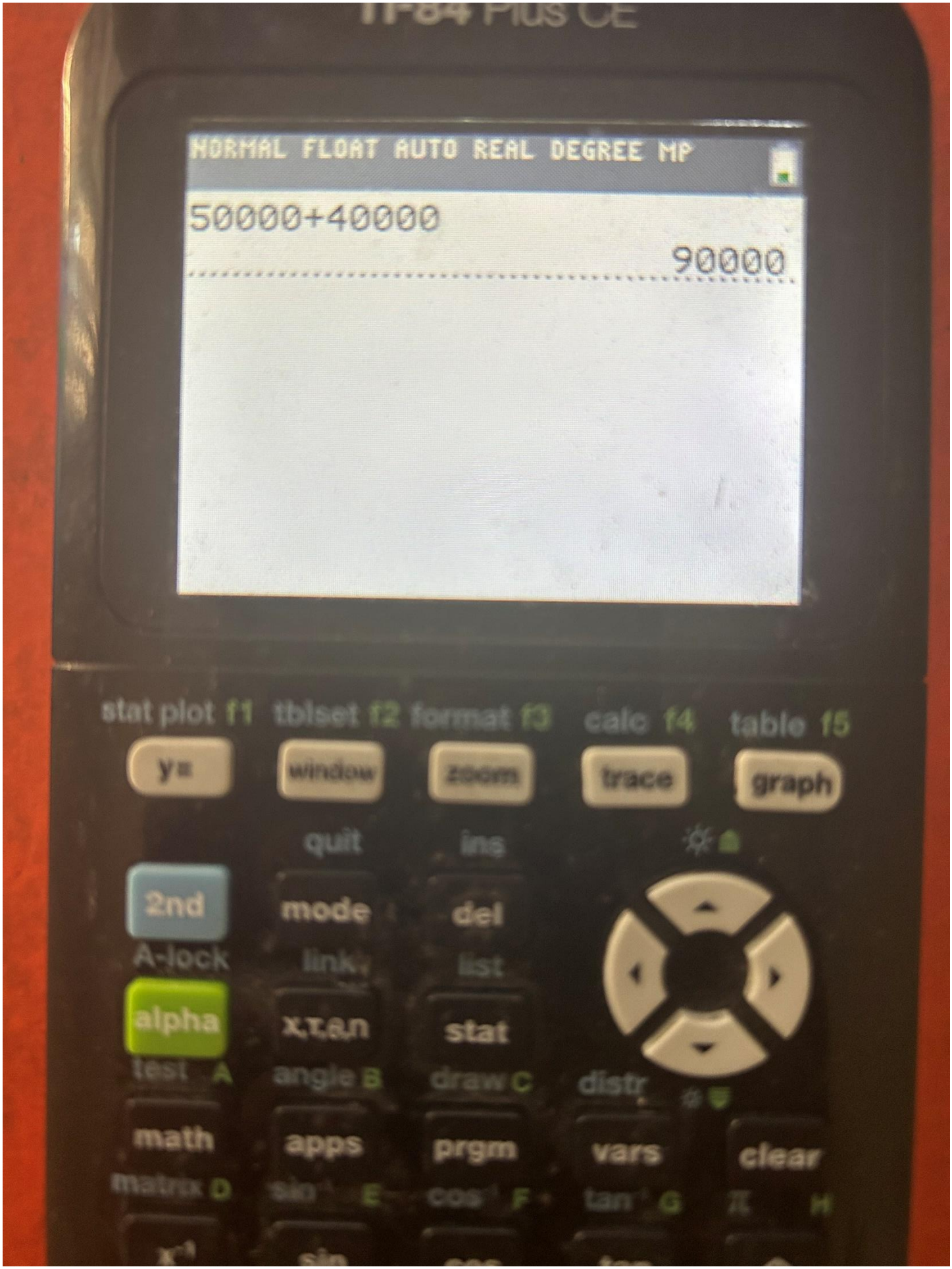
0x402002: 40000

```
 1 ; addition.asm
 2 ; unsigned short num1 = 50000;
 3 ; unsigned short num2 = 40000;
 4 ; unsigned int sum = 0;
 5 ; sum = int(num1 + num2);
 6 section .data
 7         num1 dw 50000
 8         num2 dw 40000
 9         sum  dd 0
10 section .text
11         global _start
12 _start:
13         mov     dx, 0
14         mov     ax, word[num1]                  ;al = num1 = 64h
15         add     ax, word[num2]                  ;al = al + num2 = 2Ch
16         adc     dx, 0                           ;ah = ah + 0 + CF = 01h
17         mov     word[sum+0], ax                 ;sum = al = 9ch
18         mov     word[sum+2], dx                 ;sum = ah = 0ffh
19                                                 ;mov word[sum], ax
20         Mov     rax, 60                         ;terminate excuting process
21         mov     rdi, 0                          ;exit status
22         syscall
23
24
```

```
Breakpoint 1, _start () at addition.asm:20
(gdb) x/uw &sum
0x402004:       90000
(gdb)
```

0x402004: 90000

[Insert addition simulation result verification here]


TI-84 Plus CE
NORMAL FLOAT AUTO REAL DEGREE MP
50000+40000
90000
stat plot f1
tblset f2
format f3
calc f4
table f5
y=
window
zoom
trace
graph
quit
ins
2nd
mode
del
A-lock
link
list
alpha
x,t,θ,n
stat
test A
angle B
draw C
distr
math
apps
prgm
vars
clear
matrix D
sin⁻¹ E
cos⁻¹ F
tan⁻¹ G
π H

[Insert subtraction.asm here]

```
; subtraction.asm
; signed short num1 = 20000;
; signed short num2 = 30000;
; signed int dif = 0;
; dif = int(num1 - num2)

section .data
    num1    dw  20000
    num2    dw      30000
    dif     dd      0

section .text
    global _start

_start:
    mov ax, word[num1]
    sub ax, word[num2]
    sbb dx, 0
    mov     word[dif + 0], ax
    mov     word[dif + 2], dx

    mov rax, 60
    mov rdi, 0
    syscall
```

[Insert subtraction simulation result here]

```
 1 ; subtraction.asm
 2 ; signed short num1 = 20000;
 3 ; signed short num2 = 30000;
 4 ; signed int dif = 0;
 5 ; dif = int(num1 - num2)
 6
 7 section .data
 8         num1    dw      20000
 9         num2    dw      30000
10         dif     dd      0
11
12 section .text
13         global _start
14
15 _start:
16         mov     ax, word[num1]
17         sub     ax, word[num2]
18         sbb     dx, 0
19         mov     word[dif + 0], ax
20         mov     word[dif + 2], dx
21
22         mov     rax, 60
23         mov     rdi, 0
24         syscall
```

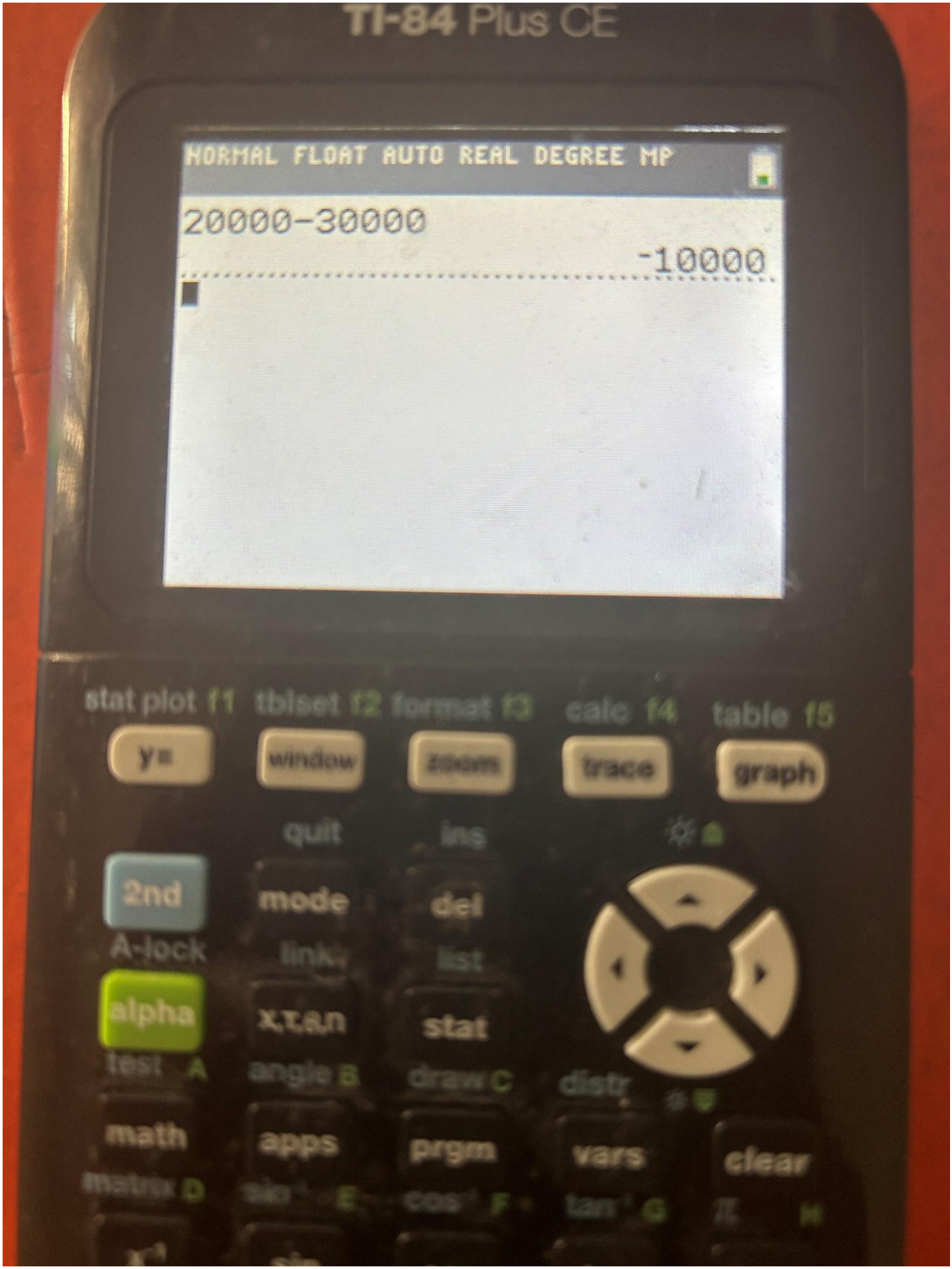
```
Breakpoint 1, _start () at subtraction.asm:22
(gdb) x/dh &num1
0x402000:       20000
(gdb) x/dh &num2
0x402002:       30000
```

```
(gdb) x/dh &num2
0x402002:       30000
(gdb) x/dw &dif
0x402004:       -10000
(gdb)
```

0x402004: -10000

0x402004: -10000

[Insert subtraction simulation result verification here]


TI-84 Plus CE
NORMAL FLOAT AUTO REAL DEGREE MP
20000-30000
-10000
stat plot f1 tblset f2 format f3 calc f4 table f5
y=
window
zoom
trace
graph
quit
ins
2nd
mode
del
A-lock
link
list
alpha
x,t,θ,n
stat
test A
angle B
draw C
distr
math
apps
prgm
vars
clear
matrix D
sin⁻¹ E
cos⁻¹ F
tan⁻¹ G
π H