

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Nguyễn Tiến Ngọc

Student code: 21522381

PRACTICE REPORT NO 5

Lab5

I. Practice content

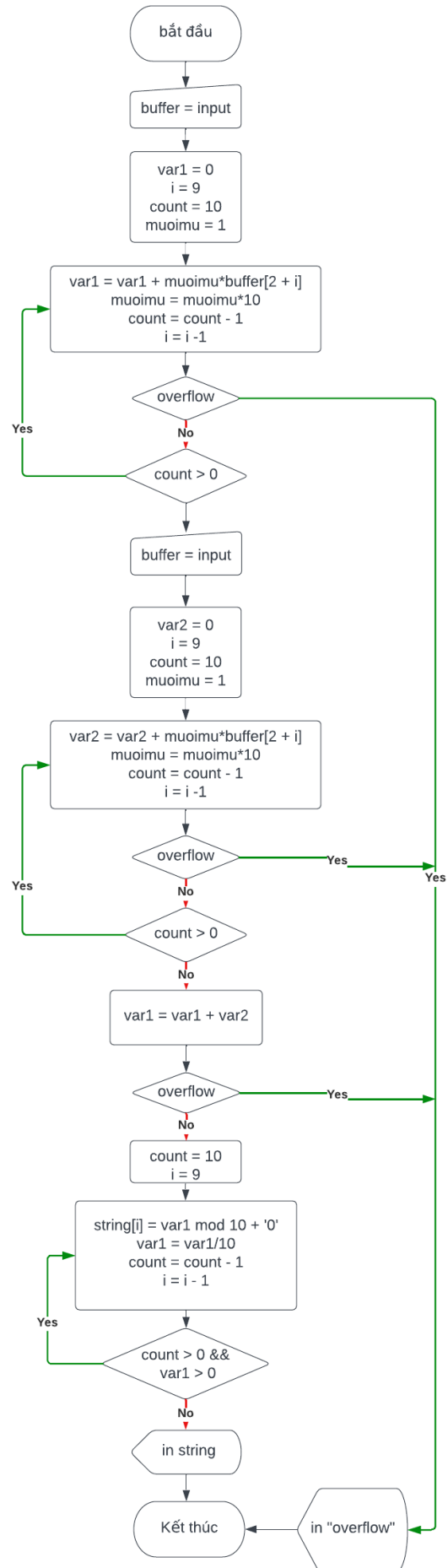
1. Example code:

```
01 name "add-sub"
02
03 org 100h
04
05 mov al, 5          ; bin=00000101b
06 mov bl, 10         ; hex=0ah or bin=00001010b
07
08 ; 5 + 10 = 15 <decimal> or hex=0fh or bin=00001111b
09 add bl, al
10
11 ; 15 - 1 = 14 <decimal> or hex=0eh or bin=00001110b
12 sub bl, 1
13
14 ; print result in binary:
15 mov cx, 8
16 print: mov ah, 2    ; print function.
17         mov dl, '0'
18         test bl, 10000000b ; test first bit.
19         jz zero
20         mov dl, '1'
21 zero:   int 21h
22         shl bl, 1
23         loop print
24
25 ; print binary suffix:
26 mov dl, 'b'
27 int 21h
28
29 ; wait for any key press:
30 mov ah, 0
31 int 16h
32
33 ret
```

- Khởi tạo giá trị ban đầu cho 2 thanh ghi al, và bl bằng lệnh mov (dòng 5, 6).
- Thực hiện cộng thanh ghi bl với thanh ghi al, kết quả lưu vào thanh ghi al.
- Thực hiện trừ thanh ghi bl đi 1, kết quả lưu vào thanh ghi bl.
- In ra kết quả ở dạng nhị phân:
 - + Dùng ngắt int 21h với ah = 2 để in ra từng kí tự.
 - + khởi tạo cho thanh ghi dl (thanh ghi chứa kí tự cần in ra) là kí tự '0'.
 - + Kiểm tra xem bit trọng số 7 có bằng 1 hay không, nếu đúng thì gán dl = '1' (kiểm tra bằng lệnh test và lệnh jz-jump if ZF=1, lệnh test thực hiện and 2 toán tử, set cờ zero nếu kết quả bằng 0, hai toán tử không thay đổi giá trị).
 - + in ra kí tự bằng lệnh int 21h với ah = 2.
 - + shift thanh ghi bl sang trái (chèn 0 vào bit có trọng số 0), sau đó lặp lại quá trình trên 8 lần (thanh ghi dl chứa 8 bit giá trị).

2. Write a program that adds 2 32-bit numbers.

a. Flowchart



b. Code

You, 2 minutes ago | 1 author (You)

```
1  ORG 100H
2  ;== NHAP VAO TOAN TU 1 DANG STRING
3      MOV DX, OFFSET BUFFER
4      MOV AH, 0AH
5      INT 21H
6
7  ;=====
8  ;TOAN TU 1, CHUYEN STRING THANH SO
9      MOV BX, OFFSET BUFFER
10     MOV CL, [BX + 1] ;DO DAI CHUOI
11     MOV CH, 0
12     MOV BX, OFFSET BUFFER + 2 ;KI TU DAU TIEN DUOC NHAP
13     MOV SI, CX ;DIA CHI CUA KI TU DANG XET(N-1 -> 0)
14     sub SI, 1
15     XULITOANTU1:
16     ;LOAD CHU SO DANG XET LEN AX
17     MOV AL, [BX + SI] ;LUU KI TU VAO AL
18     AND AH, 0 ;DAM BAO AX = AL
19     SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
20
21     ;NHAN SO DANG XET VOI WORD THAP CUA MUOIMU
22     MUL WORD PTR [0 + OFFSET MUOIMU]
23     MOV [0 + OFFSET RESULT], AX ;LUU WORD THAP CUA PHEP NHAN VAO WORD THAP CUA RESULT
24     MOV [2 + OFFSET RESULT], DX ;LUU WORD CAO CUA PHEP NHAN VAO WORD CAO CUA RESULT
25
26     ;LOAD CHU SO DANG XET LEN AX
27     MOV AL, [BX + SI] ;LUU KI TU VAO AL
28     AND AH, 0 ;DAM BAO AX = AL
29     SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
30
31     ;NHAN CHU SO DANG XET VOI WORD CAO CUA MUOIMU
32     MUL WORD PTR [2 + OFFSET MUOIMU]
33     ADD [2 + OFFSET RESULT], AX ;CONG WORD THAP CUA PHEP NHAN VAO WORD CAO CUA RESULT
34
35     ;dx chua phan bit 32-47, neu dx > 0 thi overflow vi ta chi tinh toan tren 32 bit
36     and dx, dx
37     jnz overflow
38
39     You, 2 minutes ago • xet overflow
40     ;CONG DON RESULT VAO VAR1
41     MOV AX, [0 + OFFSET RESULT]
42     CLC
43     ADD [0 + OFFSET VAR1], AX
44     mov AX, [2 + OFFSET RESULT]
45     ADC [2 + OFFSET VAR1], AX
46     JC overflow ;neu CF = 1, thi overflow
```

```

46
47 ;MUOIMU = 10*MUOIMU
48 MOV AX, 10
49 MUL word ptr [0 + OFFSET MUOIMU] ;NHAN WORD THAP CUA MUOIMU VOI 10
50 MOV [0 + OFFSET RESULT], AX
51 MOV [2 + OFFSET RESULT], DX
52 MOV AX, 10
53 MUL word ptr [2 + OFFSET MUOIMU] ;NHAN WORD CAO CUA MUOIMU VOI 10
54 ADD [2 + OFFSET RESULT], AX
55 ;BO QUA WORD CAO CUA PHEP NHAN(DX), DO TA CHI XET SO 32 BIT
56 ;NEU DX KHAC 0 THI DA TRAN SO
57
58 ;GAN MUOIMU = RESULT
59 MOV AX, [0 + OFFSET RESULT]
60 MOV [0 + OFFSET MUOIMU], AX
61 MOV AX, [2 + OFFSET RESULT]
62 MOV [2 + OFFSET MUOIMU], AX
63
64 DEC SI ; SI = SI - 1
65
66 ;CX = CX - 1, NEU CX > 0 LAP LAI QUA TRINH TREN
67 LOOP XULITOANTU1
68
69 ;IN RA KI TU XUONG DONG
70 MOV AH, 2
71 MOV DL, 0AH ;NEW LINE CHARATER
72 INT 21H
73 MOV DL, 13 ;DUA CON TRO VE DAU HANG
74 INT 21H
75
76 ;== NHAP VAO TOAN TU 2 DANG STRING
77 MOV DX, OFFSET BUFFER
78 MOV AH, 0AH
79 INT 21H
80
81 ;=====
82 ;TOAN TU 2, CHUYEN STRING THANH SO
83 MOV BX, OFFSET BUFFER
84 MOV CL, [BX + 1] ;DO DAI CHUOI
85 MOV CH, 0
86 MOV WORD PTR [0 + OFFSET MUOIMU], 1H ;MUOIMU = 1
87 MOV WORD PTR [2 + OFFSET MUOIMU], 0H
88 MOV SI, CX
89 SUB SI, 1 ;DIA CHI CUA KI TU DANG XET(N-1 -> 0)
90 MOV BX, OFFSET BUFFER + 2 ;KI TU DAU TIEN DUOC NHAP

```

```

91 XULITOANTU2:
92 ;LOAD CHU SO DANG XET LEN AX
93 MOV AL, [BX + SI] ;LUU KI TU VAO AL
94 AND AH, 0 ;DAM BAO AX = AL
95 SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
96
97 ;NHAN SO DANG XET VOI WORD THAP CUA MUOIMU
98 MUL WORD PTR [0 + OFFSET MUOIMU]
99 MOV [0 + OFFSET RESULT], AX ;LUU WORD THAP CUA PHEP NHAN VAO WORD THAP CUA RESULT
100 MOV [2 + OFFSET RESULT], DX ;LUU WORD CAO CUA PHEP NHAN VAO WORD CAO CUA RESULT
101
102 ;LOAD CHU SO DANG XET LEN AX
103 MOV AL, [BX + SI] ;LUU KI TU VAO AL
104 AND AH, 0 ;DAM BAO AX = AL
105 SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
106
107 ;NHAN CHU SO DANG XET VOI WORD CAO CUA MUOIMU
108 MUL WORD PTR [2 + OFFSET MUOIMU]
109 ADD [2 + OFFSET RESULT], AX ;CONG WORD THAP CUA PHEP NHAN VAO WORD CAO CUA RESULT
110
111 ;dx chua phan bit 32-47, neu dx > 0 thi overflow vi ta chi tinh toan tren 32 bit
112 and dx, dx
113 jnz overflow
114
115 ;CONG DON RESULT VAO VAR2
116 MOV AX, [0 + OFFSET RESULT]
117 CLC
118 ADD [0 + OFFSET VAR2], AX
119 MOV AX, [2 + OFFSET RESULT]
120 ADC [2 + OFFSET VAR2], AX
121 JC overflow ;neu CF = 1, thi overflow
122
123 ;MUOIMU = 10*MUOIMU
124 MOV AX, 10
125 MUL word ptr [0 + OFFSET MUOIMU] ;NHAN WORD THAP CUA MUOIMU VOI 10
126 MOV [0 + OFFSET RESULT], AX
127 MOV [2 + OFFSET RESULT], DX
128 MOV AX, 10
129 MUL word ptr [2 + OFFSET MUOIMU] ;NHAN WORD CAO CUA MUOIMU VOI 10
130 ADD [2 + OFFSET RESULT], AX
131 ;BO QUA WORD CAO CUA PHEP NHAN(DX), DO TA CHI XET SO 32 BIT
132 ;NEU DX KHAC 0 THI DA TRAN SO
133
134 ;GAN MUOIMU = RESULT
135 MOV AX, [0 + OFFSET RESULT]
136 MOV [0 + OFFSET MUOIMU], AX

```

```

136     MOV [0 + OFFSET MUOIMU], AX
137     MOV AX, [2 + OFFSET RESULT]
138     MOV [2 + OFFSET MUOIMU], AX
139
140     DEC SI
141     ;CX = CX - 1, NEU CX > 0 LAP LAI QUA TRINH TREN
142     LOOP XULITOANTU2
143
144     ;tinh toan var1 = var1 + var2
145     cld
146     mov AX, [0 + offset var2]
147     add [0 + offset VAR1], AX
148     mov AX, [2 + offset var2]
149     adc [2 + offset VAR1], AX
150     JC overflow ;neu CF = 1, thi overflow
151 ;=====
152     ;IN RA KI TU XUONG DONG
153     MOV AH, 2
154     MOV DL, 0AH ;NEW LINE CHARATER
155     INT 21H
156     MOV DL, 13 ;DUA CON TRO VE DAU HANG
157     INT 21H
158
159 ;=====
160     ;CHUYEN DOI SO NHI PHAN SANG STRING
161     MOV CX, 10
162     MOV SI, 9
163     CONVERT:
164     ;chia word cao cua var1 cho 10
165     MOV BX, 10
166     mov DX, 0 ;phep chia 32b/16b, nen phai dat dx = 0
167     mov AX, [2+OFFSET VAR1] ;word cao cua var1
168     DIV BX
169     MOV [2+OFFSET VAR1], AX ;WORD CAO CUA PHAN NGUYEN
170
171     ;chia word thap cua var1 cho 10
172     ;dx = phan du < 10. vi vay phep chia sau se khong bi tran
173     MOV AX, [0 + OFFSET VAR1] ;word thap cua var1
174     DIV BX
175     MOV [0+OFFSET VAR1], AX ;WORD THAP CUA PHAN NGUYEN
176     ;DX CHUA PHAN DU
177     ADD DX, '0'
178     MOV BYTE PTR [SI + OFFSET STRING], DL
179     DEC SI
180
181     ;ax chua word thap cua var1, thuc hien or word

```

```

182      ;thap va word cao de kiem tra xem var1 co bang 0 hay khong
183      ;neu bang 0 thi thoat khoi vong lap
184      mov bx, offset var1
185      or ax, [bx + 2] ;neu var1 = 0, ZF = 1
186      ;if (CX <> 0) and (ZF = 0) then jump
187      LOOPNE CONVERT
188
189
190      ;=====
191      ;in ra ket qua
192      ;sau khi thoat khoi vong lap tren, si + 1 la
193      ;dia chi cua chu so dau tien
194      inc si
195
196      mov dx, offset STRING
197      add dx, si ;tro dx den chu so dau tien khac 0
198      ;in ra chuoai, ki tu ket thuc la '$' da duoc khai bao cuoi chuoai
199      mov ah, 9
200      int 21h
201      jmp exit
202      ;=====
203      ;in ra dong chu overflow
204      overflow:
205      mov dx, offset s_overflow
206      mov ah, 9
207      int 21h
208  exit:
209      RET
210      BUFFER DB 11,?, 11 DUP(' ')
211      VAR1    DD 0
212      VAR2    DD 0
213      RESULT  DD ?
214      MUOIMU  DD 1
215      STRING  dB 10 dup('0'), '$'
216      s_overflow db 0ah, 13, "overflow $"

```

c. Explanation

* Nhập vào string, chuyển sang nhị phân:

- Khai báo 1 buffer gồm 11 kí tự, 10 chữ số và 1 kí tự kết thúc chuỗi. ($2^{32} - 1$ có thể biểu diễn bằng 10 chữ số).

- Thực hiện chuyển đổi string sang mã bcd, từ bcd chuyển sang binary bằng cách lấy từng chữ số nhân với $10^{\text{(trọng số của chữ số)}}$. Vd $432 = 4 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$.

- 8086 chỉ hỗ trợ phép nhân 16*16 bit, mà phép nhân lớn nhất ta cần thực hiện là $4 \cdot 10^9$ ($2^{32} - 1 = 4294967296$) là phép nhân 16 bit * 32 bit, kết quả là 1 số 48 bit.

+ giả sử số A 32 bit gồm phần word cao H, word thấp L1, số L2 16 bit. $A \cdot L2 = (H \cdot 2^{16} + L1) \cdot L2 = H \cdot L2 \cdot 2^{16} + H \cdot L1$. Như vậy ta đã đơn giản hóa phép nhân 32b*16 thành 16b*16 kết hợp với phép dịch (nhân 2^{16} là dịch 16 bit, nhưng thực tế trong code không dùng phép dịch mà gán kết quả của $H \cdot L2$ vào phần bit 48-16 của kết quả).

* tính toán var1 + var2:

+ lấy word thấp của var1 + word thấp của var2

+ lấy word cao của var1 + word cao của var2 + bit nhớ từ phép tính trước.

* In ra kết quả:

- chia lấy dư kết quả cho 10, gán string[i] = phần dư + '0' (i = 9 -> 0). Kết quả = kết quả / 10, i = i - 1. Lặp lại cho đến khi kết quả = 0 hoặc lặp đủ 10 lần.

+ Thực hiện phép chia 32 bit cho số 10 (16 bit): 8086 hỗ trợ phép chia 32 bit cho 16 bit, nhưng kết quả có thể bị tràn do kết quả trong phép chia chỉ được lưu ở thanh ghi 16 bit (32 bit chia 16 bit kết quả phải là 32 bit).

+ Giải sử số 32 bit có dạng H + L1, số 16 bit có dạng L2.

+ Thực hiện chia phần cao H (16 bit) cho L2: gán DX = 0 (phần cao của số 32 bit), AX = H, BX = L2. Vậy đây có thể coi như là phép chia 16 bit cho 16 bit. Kết quả là 16 bit sẽ không bị tràn. (*1)

+ Phần nguyên của phép chia 1 chính là phần cao của kết quả phép chia mong muốn (32b/16b).

+ Thực hiện chia (DX AX) cho BX: DX = phần dư của phép chia trước đó, AX = L1, BX = L2. Đây vẫn là 1 phép chia 32 bit cho 16 bit. Nhưng sẽ không bị tràn vì phần DX < L2 (vì DX là phần dư của phép chia trước, luôn bé hơn L2). Phần nguyên của phép chia này là word thấp trong phần nguyên của kết quả. Phần dư DX là phần dư của toàn bộ phép chia 32b/16b.

+ Tham khảo thêm về phép chia 32b/16b:



The method below is similar to long hand division on pencil and paper, dividing a 2 digit numerator by a single digit divisor. For example 99/4:

4



```

  2 4   (quotient)
  ----
4 | 9 9
   8
  --
   1 9
   1 6
   --
    3   (remainder)
```

Start off with dx = 0 and ax = high order of numerator. After the first div instruction, the remainder in dx is what's left of the high order of the numerator for the second divide. This method could be enhanced to handle a numerator and quotient of any size, as long as the divisor is a 16 bit value.

```
num    dd    10000000h
dvsr   dw    2000h
quot   dd    ?
rmdr   dw    ?
;
; ...
mov     cx,dvsr           ;cx = dvsr
xor     dx,dx             ;dx = 0
mov     ax,word ptr [num+2] ;ax = high order numerator
div     cx                 ;dx = rem, ax = high order quotient
mov     word ptr [quot+2],ax ;store high order quotient
mov     ax,word ptr [num]   ;ax = low order numerator
div     cx                 ;dx = rem, ax = low order quotient
mov     word ptr [quot],ax  ;store low order quotient
mov     word ptr [rmdr],dx  ;store remainder
```

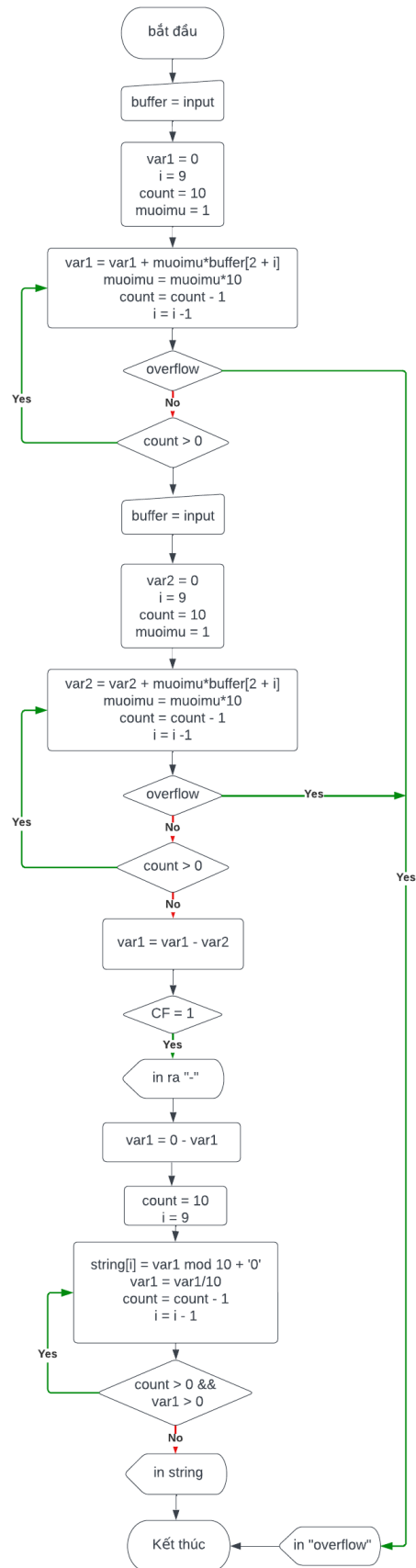

- Sau khi lặp, $i + 1$ chính là địa chỉ của số đầu tiên cần in ra, thực hiện in ra chuỗi kể từ i (ví dụ kết quả là 123 thì chỉ in ra 123 thay vì 0000000123).

d. Demonstration

- Link drive: <https://drive.google.com/file/d/1qHrkXzcZBjpND1mI-Ugg1a7SIEbp3oeO/view?usp=sharing>

III. Exercise

1. Flowchart



2. Code

```
You, 2 hours ago | 1 author (You)
1  ORG 100H
2  ;== NHAP VAO TOAN TU 1 DANG STRING      You, 2 hours ago • xet overflow ...
3      MOV DX, OFFSET BUFFER
4      MOV AH, 0AH
5      INT 21H
6
7  ;=====
8  ;TOAN TU 1, CHUYEN STRING THANH SO
9      MOV BX, OFFSET BUFFER
10     MOV CL, [BX + 1] ;DO DAI CHUOI
11     MOV CH, 0
12     MOV BX, OFFSET BUFFER + 2 ;KI TU DAU TIEN DUOC NHAP
13     MOV SI, CX ;DIA CHI CUA KI TU DANG XET(N-1 -> 0)
14     sub SI, 1
15     XULITOANTU1:
16         ;LOAD CHU SO DANG XET LEN AX
17         MOV AL, [BX + SI] ;LUU KI TU VAO AL
18         AND AH, 0 ;DAM BAO AX = AL
19         SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
20
21         ;NHAN SO DANG XET VOI WORD THAP CUA MUOIMU
22         MUL WORD PTR [0 + OFFSET MUOIMU]
23         MOV [0 + OFFSET RESULT], AX ;LUU WORD THAP CUA PHEP NHAN VAO WORD THAP CUA RESULT
24         MOV [2 + OFFSET RESULT], DX ;LUU WORD CAO CUA PHEP NHAN VAO WORD CAO CUA RESULT
25
26         ;LOAD CHU SO DANG XET LEN AX
27         MOV AL, [BX + SI] ;LUU KI TU VAO AL
28         AND AH, 0 ;DAM BAO AX = AL
29         SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
30
31         ;NHAN CHU SO DANG XET VOI WORD CAO CUA MUOIMU
32         MUL WORD PTR [2 + OFFSET MUOIMU]
33         ADD [2 + OFFSET RESULT], AX ;CONG WORD THAP CUA PHEP NHAN VAO WORD CAO CUA RESULT
34
35         ;dx chua phan bit 32-47, neu dx > 0 thi overflow vi ta chi tinh toan tren 32 bit
36         and dx, dx
37         jnz overflow
38
39         ;CONG DON RESULT VAO VAR1
40         MOV AX, [0 + OFFSET RESULT]
41         CLC
42         ADD [0 + OFFSET VAR1], AX
43         mov AX, [2 + OFFSET RESULT]
44         ADC [2 + OFFSET VAR1], AX
45         JC overflow ;neu CF = 1, thi overflow
```

```

46
47     ;MUOIMU = 10*MUOIMU
48     MOV AX, 10
49     MUL word ptr [0 + OFFSET MUOIMU] ;NHAN WORD THAP CUA MUOIMU VOI 10
50     MOV [0 + OFFSET RESULT], AX
51     MOV [2 + OFFSET RESULT], DX
52     MOV AX, 10
53     MUL word ptr [2 + OFFSET MUOIMU] ;NHAN WORD CAO CUA MUOIMU VOI 10
54     ADD [2 + OFFSET RESULT], AX
55     ;BO QUA WORD CAO CUA PHEP NHAN(DX), DO TA CHI XET SO 32 BIT
56     ;NEU DX KHAC 0 THI DA TRAN SO
57
58     ;GAN MUOIMU = RESULT
59     MOV AX, [0 + OFFSET RESULT]
60     MOV [0 + OFFSET MUOIMU], AX
61     MOV AX, [2 + OFFSET RESULT]
62     MOV [2 + OFFSET MUOIMU], AX
63
64     DEC SI ; SI = SI - 1
65
66     ;CX = CX - 1, NEU CX > 0 LAP LAI QUA TRINH TREN
67     LOOP XULITOANTU1
68
69     ;IN RA KI TU XUONG DONG
70     MOV AH, 2
71     MOV DL, 0AH ;NEW LINE CHARATER
72     INT 21H
73     MOV DL, 13 ;DUA CON TRO VE DAU HANG
74     INT 21H
75
76     ;== NHAP VAO TOAN TU 2 DANG STRING
77     MOV DX, OFFSET BUFFER
78     MOV AH, 0AH
79     INT 21H
80
81     ;=====
82     ;TOAN TU 2, CHUYEN STRING THANH SO
83     MOV BX, OFFSET BUFFER
84     MOV CL, [BX + 1] ;DO DAI CHUOI
85     MOV CH, 0
86     MOV WORD PTR [0 + OFFSET MUOIMU], 1H ;MUOIMU = 1
87     MOV WORD PTR [2 + OFFSET MUOIMU], 0H
88     MOV SI, CX
89     SUB SI, 1 ;DIA CHI CUA KI TU DANG XET(N-1 -> 0)
90     MOV BX, OFFSET BUFFER + 2 ;KI TU DAU TIEN DUOC NHAP

```

```

91 XULITOANTU2:
92 ;LOAD CHU SO DANG XET LEN AX
93 MOV AL, [BX + SI] ;LUU KI TU VAO AL
94 AND AH, 0 ;DAM BAO AX = AL
95 SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
96
97 ;NHAN SO DANG XET VOI WORD THAP CUA MUOIMU
98 MUL WORD PTR [0 + OFFSET MUOIMU]
99 MOV [0 + OFFSET RESULT], AX ;LUU WORD THAP CUA PHEP NHAN VAO WORD THAP CUA RESULT
100 MOV [2 + OFFSET RESULT], DX ;LUU WORD CAO CUA PHEP NHAN VAO WORD CAO CUA RESULT
101
102 ;LOAD CHU SO DANG XET LEN AX
103 MOV AL, [BX + SI] ;LUU KI TU VAO AL
104 AND AH, 0 ;DAM BAO AX = AL
105 SUB AX, '0' ;CHUYEN TU MA ASCII SANG DANG SO
106
107 ;NHAN CHU SO DANG XET VOI WORD CAO CUA MUOIMU
108 MUL WORD PTR [2 + OFFSET MUOIMU]
109 ADD [2 + OFFSET RESULT], AX ;CONG WORD THAP CUA PHEP NHAN VAO WORD CAO CUA RESULT
110
111 ;dx chua phan bit 32-47, neu dx > 0 thi overflow vi ta chi tinh toan tren 32 bit
112 and dx, dx
113 jnz overflow
114
115 ;CONG DON RESULT VAO VAR2
116 MOV AX, [0 + OFFSET RESULT]
117 CLC
118 ADD [0 + OFFSET VAR2], AX
119 MOV AX, [2 + OFFSET RESULT]
120 ADC [2 + OFFSET VAR2], AX
121 JC overflow ;neu CF = 1, thi overflow
122
123 ;MUOIMU = 10*MUOIMU
124 MOV AX, 10
125 MUL word ptr [0 + OFFSET MUOIMU] ;NHAN WORD THAP CUA MUOIMU VOI 10
126 MOV [0 + OFFSET RESULT], AX
127 MOV [2 + OFFSET RESULT], DX
128 MOV AX, 10
129 MUL word ptr [2 + OFFSET MUOIMU] ;NHAN WORD CAO CUA MUOIMU VOI 10
130 ADD [2 + OFFSET RESULT], AX
131 ;BO QUA WORD CAO CUA PHEP NHAN(DX), DO TA CHI XET SO 32 BIT
132 ;NEU DX KHAC 0 THI DA TRAN SO
133
134 ;GAN MUOIMU = RESULT
135 MOV AX, [0 + OFFSET RESULT]

```

```

136     MOV [0 + OFFSET MUOIMU], AX
137     MOV AX, [2 + OFFSET RESULT]
138     MOV [2 + OFFSET MUOIMU], AX
139
140     DEC SI
141     ;CX = CX - 1, NEU CX > 0 LAP LAI QUA TRINH TREN
142     LOOP XULITOANTU2
143
144     ;tinh toan var1 = var1 - var2
145     cld
146     mov AX, [0 + offset var2]
147     sub [0 + offset VAR1], AX
148     mov AX, [2 + offset var2]
149     sbb [2 + offset VAR1], AX
150
151 ;=====
152     ;IN RA KI TU XUONG DONG
153     MOV AH, 2
154     MOV DL, 0AH ;NEW LINE CHARATER
155     INT 21H
156     MOV DL, 13 ;DUA CON TRO VE DAU HANG
157     INT 21H
158     ;in ra dau tru neu kq am
159     JNC KhongInDauTru
160     MOV DL, '-'
161     INT 21H
162
163     ;var1 < 0, lay 0 - var1 de lay gia tri tuyet doi
164     CLC
165     mov AX, 0
166     MOV BX, OFFSET VAR1
167     sub AX, [BX + 0]
168     mov DX, 0
169     sbb DX, [BX + 2]
170     MOV [0 + offset VAR1], AX
171     MOV [2 + offset VAR1], DX
172     KhongInDauTru:
173
174 ;=====
175     ;CHUYEN DOI SO NHI PHAN SANG STRING
176     MOV CX, 10
177     MOV SI, 9
178     CONVERT:
179     ;chia word cao cua var1 cho 10
180     MOV BX, 10
181     mov DX, 0 ;phep chia 32b/16b, nen phai dat dx = 0

```

ASM pheptru.asm

```
182     mov AX, [2+OFFSET VAR1] ;word cao cua var1
183     DIV BX
184     MOV [2+OFFSET VAR1], AX ;WORD CAO CUA PHAN NGUYEN
185
186     ;chia word thap cua var1 cho 10
187     ;dx = phan du < 10. vi vay phep chia sau se khong bi tran
188     MOV AX, [0 + OFFSET VAR1] ;word thap cua var1
189     DIV BX
190     MOV [0+OFFSET VAR1], AX ;WORD THAP CUA PHAN NGUYEN
191     ;DX CHUA PHAN DU
192     ADD DX, '0'
193     MOV BYTE PTR [SI + OFFSET STRING], DL
194     DEC SI
195
196     ;ax chua word thap cua var1, thuc hien or word
197     ;thap va word cao dekiem tra xem var1 co bang 0 hay khong
198     ;neu bang 0 thi thoat khoi vong lap
199     mov bx, offset var1
200     or ax, [bx + 2] ;neu var1 = 0, ZF = 1
201     ;if (CX <> 0) and (ZF = 0) then jump
202     LOOPNE CONVERT
203
204
205     ;=====
206     ;in ra ket qua
207     ;sau khi thoat khoi vong lap tren, si + 1 la
208     ;dia chi cua chu so dau tien
209     inc si
210
211     mov dx, offset STRING
212     add dx, si ;tro dx den chu so dau tien khac 0
213     ;in ra chuoi, ki tu ket thuc la '$' da duoc khai bao cuoi chuoi
214     mov ah, 9
215     int 21h
216     jmp exit
217     ;=====
218     ;in ra dong chu overflow
219     overflow:
220     mov dx, offset s_overflow
221     mov ah, 9
222     int 21h
223 exit:
224     RET
225     BUFFER DB 11,?, 11 DUP(' ')
226     VAR1    DD 0
227     VAR2    DD 0
```

```
228     RESULT DD ?
229     MUOIMU DD 1
230     STRING db 10 dup('0'), '$'
231     s_overflow db 0ah, 13, "overflow $"
```

3. Explanation

- Mọi thứ giống như bài phép cộng, chỉ khác là phép trừ không xét overflow ở kết quả (2 số dương trừ nhau không thể overflow).
- Xét xem kết quả có âm hay không bằng cờ CF, nếu âm thì in ra dấu trừ và kết quả = 0 – kết quả (lấy giá trị tuyệt đối).

4. Demonstration

- Link Drive: <https://drive.google.com/file/d/1UpLTkgakvgzOD4B6Qxqb6xHLfh6ePYx1/view?usp=sharing>