

### Question 1

external atoi

```
%macro print_num 1
    mov eax,0
    mov dword[NUMBER], %1
    push NUMBER
    call atoi
    pop eax        //can pop to mem ?
    mov dword[NUMBER], eax    //what if number is too big??
    mov eax,0x04            //sys call number 4: print
    mov ebx,0x01            //stdout
    mov ecx, NUMBER        //buffer
    mov edx 0x020          //32 decimal bytes to write
    int      0x80          //invoke sys
    //can't use labels to error printing
%endmacro
```

### Question 2

No answer

The code checks how many ones are in ebx

?? there must be a test whether we searched the whole ebx

Byte masking

Calling func?

Doing 32 times

### Question 3.1

```
mov ax, 0x01
inc ax        //assuming ax is 0
add ax, 0x01  //assuming ax is 0
or ax, 0x01   //assuming ax is 0
```

### Question 3.2

No answer

STC, Shr ??

?? not good not sure the other digits are zero

#### Q4

No assumption that there be no overflow.

No answer

// how there would be an overflow???

Assume there's no over flow:

//over flow exists in sub as well ???

5 binary = 00000101

2 binary = 00000010 => 2 (2's compl) binary = 11111110

In 2 bytes Big E.= 00000000.11111110

In 2 bytes Little E.= 11111110.00000000

//how do I know to read this 1111?

1 binary = 00000001

a. For intel

To add => 11111110.00000000

Little Endian => 00000101.11111110.00000000.00000001

ADD x, 00000101111111100000000000000001

// got confused where to add to starting x to the right ? or z to the left?

00000001	01001010	0000001	0000111
00000101	11111110	00000000	00000001

b. For motorola

To add => 00000101.00000000.11111110.00000001

ADDI.L 00000101000000001111111000000001, x

00000001	0000001	01001010	0000111
00000101	00000000	11111110	00000001

#### Q5.1

//note: the carry flag know to "adjust" itself. It knows that it is the carry of 2 numbers of 16 bytes (I thought that it is only the carry for sum of 2 numbers of 32 bytes each).

//assuming the CF is initialized to 0

Output:

- Adds  $AX \oplus BX$  CX times
- Saves the sum of CX carry bits that raised from the CX steps of the calculation.

Q5.2

No answer

??

שאלה 6.

No answer

מחסירים מ D2 את D0 K פעמים.

שאלה 7.

No answer

לא מבין למה הטבלה? מערך קונבנציות?

שאלה 8

נתונה מכונה אחת. מהירותה  $25 * 10^8 \frac{C}{S}$ . מריצים עליה תוכנית ש 10% מזמנה הוא INT ; 30% הוא Float ; 60% גרפי. אם נשפר את הגרפי להיות פי X מהיר יותר נגרום לfloat להיות פי  $\frac{x}{4}$  איטי יותר. מהו X מקמסלי?  
שיפור גרפי אומר פחות מחזורים.

$$et(x) = 10\% * old\_et + \frac{30\% * old\_et}{\frac{4}{x}} + \frac{60\% * old\_et}{x}$$

נגזור לפי X, אח"כ נשווה ל0:

$$\frac{d}{dx} \left( 10\% * old\_et + \frac{30\% * old\_et}{\frac{4}{x}} + \frac{60\% * old\_et}{x} \right) = \frac{30\% * old\_et}{4} - \frac{60\% * old\_et}{x^2} = 0$$

$$x = \sqrt{8} \quad \frac{1}{4} = \frac{2}{x^2}$$

לא השתמשתי בכלל במהירות שלה ???

שאלה 9

נתונה מכונה אחת. מהירות מהירותה  $25 * 10^8 \frac{C}{S}$ . 3 קב' קוד. G1 לוקח 3 מחזורים. G2 לוקח 4 מחזורים. G3 לוקח 6 מחזורים.  
שתי תוכניות עם אותו אורך קוד. קוד של 1 מכיל 80% G1, 10% G2, 10% G3. קוד של 2 מכיל 30% G1, 40% G2, 30% G3. איזה קוד ירוץ יותר מהר (מהירות של קוד = כמה מחזורים הוא לוקח. כי זו אותה מכונה. גם פה מהירות המכונה לא רלבנטית).

קוד 1 – פחות מחזורים. מהיר יותר.

#### שאלה 10

1. Hamming Distance = 6
2. יכול לזהות  $d-1=5$  שגיאות
3. יכול לתקן  $\left\lfloor \frac{d-1}{2} \right\rfloor = 2$  שגיאות (מעוגל למטה)
4. יכול לתקן  $d-1=5$  מחיקות