Lab 6

**Name:** Vaghani Smit Dhirubhai

**Roll No:** 166

**College Id:** 19CEUEG022

1. **Write the programs to verify the instructions AAA, AAS, AAM, AAD, DAA and DAS instructions.**

**Code:**

code *segment*

*assume* *cs*:*code*

    mov ah,0

    mov *al*,'6'

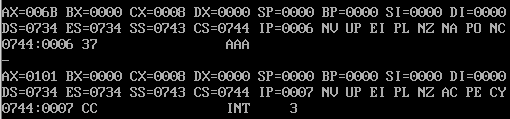
    add *al*,'5'

    aaa

    int 03

code ends

end

**Screenshot:**

code *segment*

*assume* *cs*:*code*

    mov ah,0

    mov *al*,'3'

    sub *al*,'9'

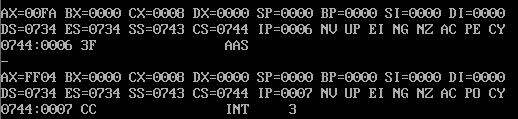
    aas

    int 03

code ends

end

**Screenshot:**

****

code *segment*

*assume* *cs*:*code*

    mov ah,0

    mov *al*,'2'

    mov *bl*,'7'

    and *al*,0fh

    and *bl*,0fh

    mul *bl*

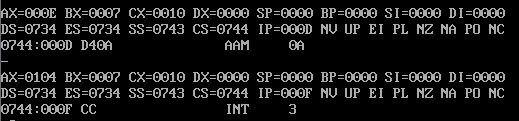
    aam

    int 03

code ends

end

**Screenshot:**

****

code *segment*

*assume* *cs*:*code*

    mov *ax*,0302h

    aad

    mov *bl*,5

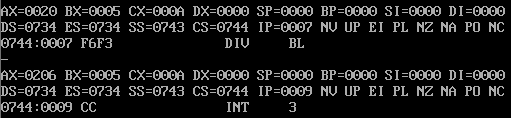
    div *bl*

    int 03

code ends

end

**Screenshot:**

****

code *segment*

*assume* *cs*:*code*

    mov *al*,34h

    add *al*,48h

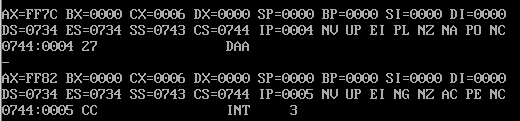
    daa

    int 03

code ends

end

**Screenshot:**

****

code *segment*

*assume* *cs*:*code*

    mov *al*,34h

    sub *al*,18h

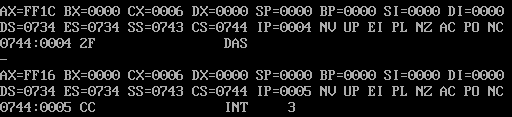
    das

    int 03

code ends

end

**Screenshot:**

****

1. **Implement a calculator for single digit numbers, which takes the input through the keyboard and display the result on the screen.**

**Code:**

code *segment*

*assume* *cs*:*code*

    xor *ax*,*ax*

    mov ah,1*;first operand*

    int 21h

    mov *bl*,*al*

    xor *ax*,*ax*

    mov ah,1*;operation*

    int 21h

    mov *cl*,*al*

    xor *ax*,*ax*

    mov ah,1*;second operand*

    int 21h

    mov bh,*al*

    cmp *cl*,'+'

    jz *addition*

    cmp *cl*,'-'

    jz *subtraction*

    cmp *cl*,'\*'

    jz *multiplication*

    cmp *cl*,'/'

    jz *division*

addition:

    xor *ax*,*ax*

    mov *al*,bh

    add *al*,*bl*

    aaa

    or *ax*,3030h

    mov *bx*,*ax*

    mov *dl*,0ah *;new line*

    mov ah,2

    int 21h

    mov *dl*,bh

    mov ah,2

    int 21h

    mov *dl*,*bl*

    mov ah,2

    int 21h

    jmp *exit*

subtraction:

    xor *ax*,*ax*

    mov *al*,*bl*

    sub *al*,bh

    aas

    or *ax*,3030h

    mov *bx*,*ax*

    mov *dl*,0ah *;new line*

    mov ah,2

    int 21h

    mov *dx*,*bx*

    mov ah,2

    int 21h

    jmp *exit*

multiplication:

    xor *ax*,*ax*

    and *bx*,0f0fh

    mov *al*,bh

    mul *bl*

    aam

    or *ax*,3030h

    mov *bx*,*ax*

    mov *dl*,0ah *;new line*

    mov ah,2

    int 21h

    mov *dl*,bh

    mov ah,2

    int 21h

    mov *dl*,*bl*

    mov ah,2

    int 21h

    jmp *exit*

division:

    xor *ax*,*ax*

    and *bx*,0f0fh

    mov *al*,*bl*

    div bh

    or *ax*,3030h

    mov *bx*,*ax*

    mov *dl*,0ah *;new line*

    mov ah,2

    int 21h

    mov *dx*,*bx*

    mov ah,2

    int 21h

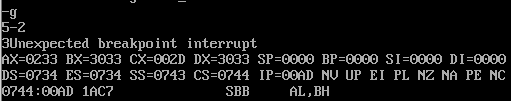
    jmp *exit*

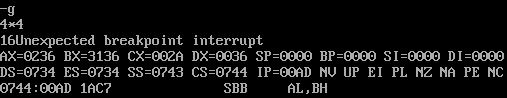
exit:*int* 3

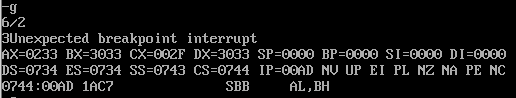
code ends

end

**Screenshots:**

****

****

****

1. **Write a program to concatenate two strings STR1 and STR2 and store the result in the string STR3. Try to input the string through the keyboard.**

**Code:**

data *segment*

*ins1* db "Enter First String: ",'$'

*ins2* db "Enter Second String: ",'$'

*ins3* db "Concated String: ",'$'

*str1* db 25,?,25 dup(0)

*str2* db 25,?,25 dup(0)

*str3* db ?

data ends

code *segment*

*assume* *cs*:*code*,*ds*:*data*

 mov *ax*,*data*

 mov ds,*ax*

 mov *es*,*ax*

*;print first string*

 mov ah,09h

 lea *dx*,*ins1*

 int 21h

 lea *dx*,*str1*

 mov ah,0ah

 int 21h

 xor *bx*,*bx* *;put terminate character*

 mov *bl*,*str1*[1]

 mov *str1*[*bx*+02h],'$'

 mov *dl*,0ah *;new line*

 mov ah,02h

 int 21h

*;print second string*

 mov ah,09h

 lea *dx*,*ins2*

 int 21h

 lea *dx*,*str2*

 mov ah,0ah

 int 21h

 xor *bx*,*bx* *;put terminate character*

 mov *bl*,*str2*[1]

 mov *str2*[*bx*+02h],'$'

 mov *dl*,0ah *;new line*

 mov ah,02h

 int 21h

 lea *dx*,*ins3*

 mov ah,09h

 int 21h

 lea *si*,*str1*[2]

 lea *di*,*str3*

 xor *cx*,*cx*

 mov *cl*,*str1*[1]

 cld

 rep *movsb*

 lea *si*,*str2*[2]

 xor *cx*,*cx*

 mov *cl*,*str2*[1]

 cld

 rep *movsb*

 mov byte *ptr* [*di*],'$'

 lea *dx*,*str3*

 mov ah,09h

 int 21h

*;newline*

 mov *dl*,0ah

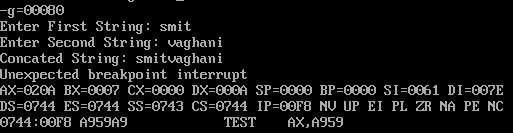
 mov ah,02h

 int 21h

 int 03

code ends

end

**Screenshot:**