Name: Amber Khurshid

Section: BAI-4A

Roll No: 22P-9295

COAL ASSIGNMENT 2

Subroutines:

1. Make separate subroutines for add, subtract, multiply and divide and then perform all these operations between two numbers of your choice using these subroutines and passing them the numbers as parameters on the stack. Also, store the results for each of the operations in the variables shown in the starter code.

```
[org 0x0100]
operand1: dw 0
operand2: dw 0
sum_result: dw 0
subtraction_result: dw 0
multiplication_result: dw 0
division_result: dw 0

push_in_stack:
pop dx
mov ax, [operand1]
push ax
mov ax, [operand2]
push ax
push dx
ret
```

```
addition:
pop dx
pop ax
pop bx
add ax, bx
push ax
push dx
ret
subtract:
pop dx
pop ax
pop bx
sub bx, ax
push bx
push dx
ret
multiply:
pop dx
pop bx
pop ax
push dx
mul bx
pop dx
```

```
push ax
push dx
ret
divide:
pop dx
pop bx
pop ax
push dx
xor dx,dx
idiv bx
pop dx
push ax
push dx
ret
start:
call push_in_stack
call addition
pop ax
mov [sum_result], ax
call push_in_stack
call subtract
pop ax
mov [subtraction_result], ax
```

```
call push_in_stack
call multiply
pop ax
mov [multiplication_result], ax
call push_in_stack
call divide
pop ax
mov [division_result], ax
```

ax, 0x4c00

int 0x21

2. Perform recursion in assembly language using subroutines of your choice.

```
[org 0x0100]

jmp start
data:   dw 10
count:  dw 2
```

```
result: dw 0


rec:
        push bp
        mov bp,sp
        mov ax,[bp+4]
        add [result],ax
        dec word[count]
        cmp word[count], 0
        jle end_rec

        push word[result]
        push word[count]
        call rec
        add sp,4
 end_rec:
        pop bp
        ret
start:  mov ax,[data]
        push ax
        mov cx,[count]
        push cx
        call rec
```

```
mov ax, 0x4c00

int 0x21
```