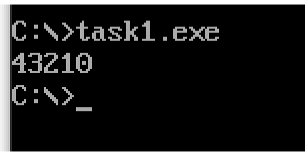
COAL LAB 7
ABDUL SAMI QASIM
22I-1725

Task1:
Code:

```
.model small
.stack 100h
.data
arr1 db 0, 1, 2, 3,4
arr2 db 5 dup(?)
.code
reverse proc
push bp
mov bp,sp
add bp,4
mov si,[bp]
add si, 5
sub si,1
mov di,offset arr2
mov cx,5
l1:
mov al,[si]
mov [di],al
add di,1
sub si,1
loop l1
pop bp
ret 2
reverse endp
```

```
main proc
mov ax, @data
mov ds, ax
mov si,OFFSET arr1
push si
call reverse
mov si,OFFSET arr2
mov cx,5
l2:
mov dl,[si]
add dl,30h
mov ah,02h
int 21h
add si,1
loop l2
mov ah,4ch
int 21h
main endp
end main
```

Output:



```
C:\>task1.exe
43210
C:\>_
```

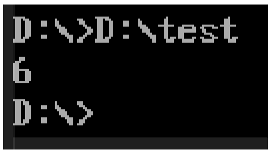
Task2 :

Code :

```
..model small
.stack 100h
.data
   num1   dw 0
   num2   dw 0
```

```
    num3   dw 0
    result dw 0

.code
main proc
                mov  ax, @data
                mov  ds, ax

    ; Input numbers
                mov  ax, 1
                mov  num1, ax

                mov  ax, 1
                mov  num2, ax

                mov  ax, 2
                mov  num3, ax

    ; Call procedure to calculate sum of squares
                push num3
                push num2
                push num1
                call sum_of_squares

    ; Retrieve result
                mov  ax, result
                mov  ax, result
                mov  dx, ax
                add  dl,'0'
                mov  ah,02h
                int  21h



                mov  ax, 4c00h
                int  21h
main endp

sum_of_squares proc
                push bp
                mov  bp, sp

                mov  ax, [bp+4]
                mul  ax
                Mov  result, ax
                Mov  ax, [bp+6]
                Mul  ax
                Add  result, ax
                Mov  ax, [bp+8]
                Mul  ax
                Add  result ,ax


                pop  bp
                ret  6
sum_of_squares endp

End main
```

Output:



```
D:\>D:\test
6
D:\>
```

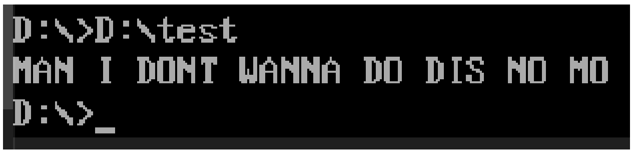
Task3:

Code:

```
.model small
.stack 100h

.data
   inputString db "man i dont wanna do dis no mo$"

.code

convertToUpper proc
   push bp
   mov bp, sp

   mov si, [bp+4]  ; Load the address of the input string from the stack

convertLoop:
   mov al, [si]    ; Load a character from the input string
   cmp al, '$'     ; Check if it's the end of the string
   je convertEnd

   cmp al, 'a'     ; Check if the character is lowercase
   jb convertNext
   cmp al, 'z'
   ja convertNext

   sub al, 32      ; Convert lowercase to uppercase by subtracting 32
   mov [si], al    ; Store the uppercase character back in the input string

convertNext:
   inc si         ; Move to the next character
   jmp convertLoop ; Repeat the loop

convertEnd:
   pop bp
   ret 2          ; Remove the return address and input string address from the stack

convertToUpper endp

main proc
   mov ax, @data
   mov ds, ax

   lea si, inputString

   push si        ; Push the address of the input string onto the stack
   call convertToUpper
   pop si         ; Remove the input string address from the stack
```

```
    ; Print the converted string
    mov ah, 09h
    mov dx, offset inputString
    int 21h

    mov ah, 4ch
    int 21h

main endp

end main
```

Output:



```
D:\>D:\test
MAN I DONT WANNA DO DIS NO MO
D:\>_
```