TASK 1
.model small

.STACK

.DATA
.code
mov cx,10
mov dx,48
l1:
mov ah,02
int 21h
inc dx
loop l1

mov dx,57
mov cx,10

l2:

mov ah,02
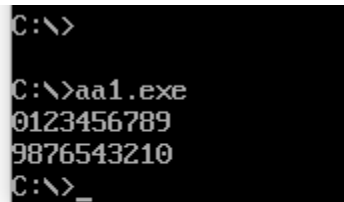int 21h
dec dx
loop l2
MOV AH,4ch

INT 21h

END


```
C:\>

C:\>aa1.exe
0123456789
9876543210
C:\>_
```
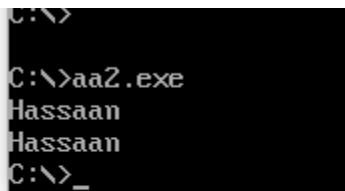
TASK 2:
.model small

.STACK

.DATA

```
str1 DB 'Hassaan$'
arr db 'H','a','s','s','a','a','n'

.CODE
MOV AX,@DATA
MOV DS,AX
mov dx,offset str1
mov ah,09
int 21h


mov dl,10
mov ah,02h
int 21h

mov si,offset arr
mov cx,7
mov dx,[si]
l1:
mov ah,02
int 21h
inc si
mov dx,[si]
loop l1


MOV AH,4ch
INT 21h
END
```



```
C:\>
C:\>aa2.exe
Hassaan
Hassaan
C:\>_
```

TASK 3:
```
.model small

.STACK

.DATA
```

```
str1 DB 'Hassaan$'
.CODE
MOV AX,@DATA
MOV DS,AX

mov si,offset str1
mov cx,7
l1:
mov dx,[si]
push dx
inc si
loop l1
mov cx,7
l2:
pop dx
mov ah,02
int 21h
loop l2

MOV AH,4ch

INT 21h

END
```



```
C:\>
C:\>aa3.exe
maassaH
C:\>_
```

TASK 4:
```
.model small

.STACK

.DATA
digi db 0
antc db 0
entc db 0
entn dw 0
temp1 dw 0
.code
MOV AX,@DATA
MOV DS,AX
input:
```

```asm
        mov ah,01h
        int 21h
        cmp al,13
        je here
        sub al,48
        mov ah,0
        mov temp1,ax
        mov ax,0
        mov ax,entn
        mov bl,10
        mul bl
        add ax,temp1
        mov entn,ax
        inc digi
        loop input
here:
        mov bx,entn

        cmp bx,80
        jae labA
        cmp bx,76
        jae labB
        cmp bx,71
        jae labC
        cmp bx,66
        jae labD
        jb labF

        jmp labB


labA:
        mov dl,'A'
        mov ah,02h
        int 21h
        jmp exit
labB:
        mov dl,'B'
        mov ah,02h
        int 21h
        jmp exit
labC:
        mov dl,'C'
        mov ah,02h
        int 21h
        jmp exit
```
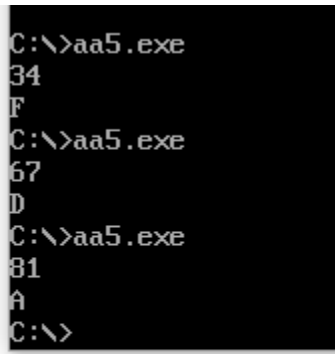
```
labD:
mov dl,'D'
mov ah,02h
int 21h
jmp exit
labF:
mov dl,'F'
mov ah,02h
int 21h
jmp exit
exit:
mov ah, 4ch
int 21h
end
```



```
.model small
.stack 100h
.data

msg1 db 'Enter two digit number : $'
msg2 db 'even$'
msg3 db 'odd$'

digitCount db 0
enteredNumber dw 0
temp1 dw 0

.code

mov ax,@data
mov ds,ax

mov ax,0
mov bx,0
mov cx,0
```

```
        mov dx,0

        lea dx, msg1
        mov ah,09h
        int 21h

        Input:

        mov ah,01
        int 21H

        cmp al,13
        JE output

        sub al,48
        mov ah,0

        mov temp1,ax

        mov ax,0

        mov ax,enteredNumber

        mov bl,10

        mul bl
        add ax,temp1

        mov enteredNumber,ax
        inc digitCount

        jmp Input

        output:

        mov ax , enteredNumber
        mov bl , 2
        div bl

        mov al, ah
        cmp al , 0
        je nume
        jmp numo

        nume:
```

```asm
mov dx, offset msg2
mov ah, 09h
int 21h

jmp exit

numo:

mov dx, offset msg3
mov ah, 09h
int 21h

jmp exit

exit:

mov ah , 4ch
int 21h
```
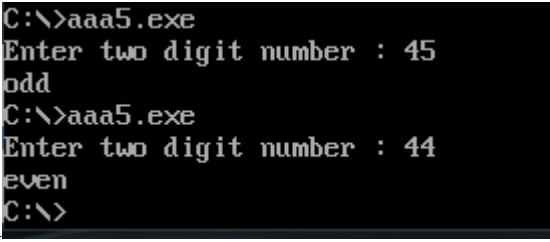
```
end
```