Lab final
Rukonuzzaman Topu

1

```
.MODEL SMALL
.STACK 100H
.DATA
NUM DB ?
STR DB "ENTER THE NUMBER: $"
STR1 DB "RESULT: $"
 NEWLINE DB 13, 10, '$'
.CODE

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    LEA DX, STR
    MOV AH,9
    INT 21H

    MOV AH,1
    INT 21H
    SUB AL,48
    MOV NUM,AL


    CMP NUM,5
    JB MULTIPLICATION
    JAE END_CODE

    MULTIPLICATION:
    MOV AL,NUM
    MOV BL,2
    MUL BL
    MOV NUM,AL

    END_CODE:

    LEA DX,NEWLINE
    MOV AH,9
```

```
    INT 21H

    LEA DX, STR1
    MOV AH,9
    INT 21H

    MOV DL,NUM
    ADD DL,48
    MOV AH,2
    INT 21H




    MOV AH,4CH
    INT 21H

    MAIN ENDP
END MAIN
```

2

```
.model small
.stack 100h

.data
    str1 db "How many numbers (1-9): $"
    str2 db 0Dh,0Ah, "Enter number: $"
    str3 db 0Dh,0Ah, "Sum of Even numbers: $"
    str4 db 0Dh,0Ah, "Sum of Odd numbers: $"

    arr db 20 dup(?)

    n    db ?
    even db 0
    odd  db 0

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

```
    mov ah, 9h
    mov dx, offset str1
    int 21h

    mov ah, 1h
    int 21h
    sub al, 30h
    mov n, al


    mov si, 0
    mov cx, 0
    mov cl, n

input_loop:
    mov ah, 9h
    mov dx, offset str2
    int 21h

    mov ah, 1h
    int 21h
    sub al, 30h

    mov arr[si], al
    inc si
    loop input_loop

    ; initialize sums
    mov si, 0
    mov cx, 0
    mov cl, n
    mov even, 0
    mov odd, 0

sum_loop:
    mov al, arr[si]
    inc si

    mov bl, al
    and al, 1
    jz even_num
```

```
    add odd, bl
    jmp next_num

even_num:
    add even, bl

next_num:
    loop sum_loop

    ; print even sum
    mov ah, 9h
    mov dx, offset str3
    int 21h

    mov al, even
    call print_2digit

    ; print odd sum
    mov ah, 9h
    mov dx, offset str4
    int 21h

    mov al, odd
    call print_2digit


    mov ah, 4Ch
    int 21h
main endp


; Print 2-digit number


print_2digit proc
    xor ah, ah
    mov bl, 10
    div bl

    mov bh, ah

    add al, 30h
    mov dl, al
    mov ah, 2h
```

```
    int 21h

    mov al, bh
    add al, 30h
    mov dl, al
    mov ah, 2h
    int 21h

    ret
print_2digit endp

end main
```

3

```
PRINT MACRO MSG
    MOV DX, OFFSET MSG
    MOV AH, 9
    INT 21H
ENDM


SQU_NUM MACRO A, RESULT
    MOV AL, A
    MUL AL
    MOV RESULT, AL
ENDM


.MODEL SMALL
.STACK 100H

.DATA
MSG1     DB "Enter your number (0-9): $"
MSG_SQU DB "SQUARE: $"

NUM1 DB ?
RES  DB ?

.CODE
MAIN PROC
    MOV AX, @DATA
```

```
MOV DS, AX


PRINT MSG1
CALL NEWLINE


MOV AH, 1
INT 21H
SUB AL, 30H
MOV NUM1, AL

CALL NEWLINE


SQU_NUM NUM1, RES


PRINT MSG_SQU


MOV AL, RES
MOV AH, 0
MOV BL, 10
DIV BL

MOV BH, AH


ADD AL, 30H
MOV DL, AL
MOV AH, 2
INT 21H


MOV AL, BH
ADD AL, 30H
MOV DL, AL
MOV AH, 2
INT 21H
```

```
    CALL NEWLINE


    MOV AH, 4CH
    INT 21H
MAIN ENDP


NEWLINE PROC
    MOV AH, 2
    MOV DL, 13
    INT 21H
    MOV DL, 10
    INT 21H
    RET
NEWLINE ENDP

END MAIN
```