

**Assignment 1 ON**  
**Assembly Language**  
**(Microprocessor – CSE 416)**

**Submitted by:**

**Name : Md. Sabbir Hasan Bhuiyan**

**ID: 20701040**

**Semester : 4<sup>th</sup>**

**Session : 2019-2020**

**Date : 14-02-2023**

**Question1:** String input and String Output

```
msg db "Enter a string: ", 0
```

```
buffer db "empty buffer --- empty buffer"
```

```
buf_size equ $ - offset buffer
```

```
org 100h
```

```
jmp start
```

```
get_string proc near
```

```
push ax
```

```
push cx
```

```
push di
```

```
push dx
```

```
mov cx, 0
```

```
cmp dx, 1
```

```
jbe exit_get_string
```

```
dec dx
```

```
wait_for_key:
```

```
mov ah, 0
```

```
int 16h
```

```
cmp al, 0Dh
jz exit_get_string
cmp al, 8
jne add_to_buffer
jcxz wait_for_key
dec cx
dec di
putc 8
putc ''
putc 8
jmp wait_for_key
```

```
add_to_buffer:
cmp cx, dx
jae wait_for_key
mov [di], al
inc di
inc cx
mov ah, 0eh
int 10h
jmp wait_for_key
```

```
exit_get_string:
mov [di], 0
```

pop dx

pop di

pop cx

pop ax

ret

get\_string endp

print\_string proc near

push ax

push si

next\_char:

mov al, [si]

cmp al, 0

jz printed

inc si

mov ah, 0eh

int 10h

jmp next\_char

printed:

pop si

pop ax

ret

print\_string endp

start:

lea si, msg

call print\_string

lea di, buffer

mov dx, buf\_size

call get\_string

putc 0Dh

putc 10

mov si, di

call print\_string

mov ax, 0

int 16h

ret

putc macro char

push ax

mov al, char

mov ah, 0eh

int 10h

pop ax

endm

print macro sdat

local next\_char, s\_dcl, printed, skip\_dcl

push ax

push si

jmp skip\_dcl

s\_dcl db sdat, 0

skip\_dcl:

lea si, s\_dcl

next\_char:

mov al, cs:[si]

cmp al, 0

jz printed

inc si

mov ah, 0eh

int 10h

jmp next\_char

printed:

pop si

pop ax

endm

printn macro sdat

```
local next_char, s_dcl, printed, skip_dcl
```

```
push ax
```

```
push si
```

```
jmp skip_dcl
```

```
s_dcl db sdat, 0Dh, 10, 0
```

```
skip_dcl:
```

```
lea si, s_dcl
```

```
next_char:
```

```
mov al, cs:[si]
```

```
cmp al, 0
```

```
jz printed
```

```
inc si
```

```
mov ah, 0eh
```

```
int 10h
```

```
jmp next_char
```

```
printed:
```

```
pop si
```

```
pop ax
```

```
endm
```

**Question 2:** Printing series 1 2 3 4 5 .....

```
.MODEL SMALL
```

```
└.STACK 100H
```

```
.DATA
```

```
upperLimit DW ? ; the upper limit of the sum
```

```
sum DW 0 ; the sum of the numbers from 1 to upperLimit
```

```
msg1 DB "Enter the range of the series: $"

msg2 DB 0DH,0AH,"The sum of the series is: $"

digit DB 0 ; a digit to be printed

.CODE

.STARTUP

MOV AX, @DATA ; Initialize DS register

MOV DS, AX

LEA DX, msg1 ; Load the address of msg1 into DX

MOV AH, 9 ; Print the string stored in msg1

INT 21H

MOV AH, 1 ; Read input from the keyboard

INT 21H

SUB AL, 30H ; Convert the ASCII code to binary

MOV upperLimit, AX ; Store the input in upperLimit

MOV AX, 0 ; Initialize AX to 0

MOV BX, 1 ; Initialize BX to 1

AGAIN: ADD AX, BX ; Add BX to AX

INC BX ; Increment BX

CMP BX, upperLimit ; Compare BX to upperLimit

JLE AGAIN ; Repeat the loop if BX <= upperLimit

MOV sum, AX ; Store the sum in sum

LEA DX, msg2 ; Load the address of msg2 into DX

MOV AH, 9 ; Print the string stored in msg2

INT 21H

MOV AX, sum
```