**מעבדה 6**

**מגישים:**

טל יהושע, ת"ז 314774860

משה פרץ, ת"ז 318263977

**תרגיל מס' 1:**

**תוכנית:**

;

; This program give access to the user to use the Stack and finding the minimum and maximum in the current time in stack

;

.MODEL SMALL

.STACK 100h

.DATA

Menu DB 13,10,'Enter 1.push 2.pop 3.min 4.max 5.exit',13,10,'$'

Buffer DW 0 ; buffer represent how much organs we push to stack

StackF DB 13,10,'Stack is FULL',13,10,'$'

StackE DB 13,10,'Stack is empty',13,10,'$'

WrongInput DB 13,10,'Wrong input, please try again',13,10,'$'

Getdown DB 13,10,13,10,'$'

.CODE

.386

Dos\_Print PROC NEAR ; set the value from AL Reg' in DL Reg' and print it

MOV DL, AL

MOV AH, 2

INT 21h

RET

Dos\_Print ENDP

Print\_String PROC NEAR     ; Help us print stuff, we need to set in DX Reg' offset first.

MOV AH, 9

INT 21h

RET

Print\_String ENDP

Get\_Num PROC NEAR ; Get the option from the user and check if its currect

MOV AH, 1h

INT 21h

XOR AH, AH

MOV DX, OFFSET Getdown

CALL Print\_String

CMP AL, '5' ; check if the input is currect

JA WrongInp

CMP AL, '1'

JL WrongInp

RET

Get\_Num ENDP

Get\_Value PROC NEAR ; Get value from the user and check if its between 0-9

MOV AH, 1h

INT 21h

XOR AH, AH

CMP AL, '9'

JA WrongInp

CMP AL, '0'

JL WrongInp

RET

Get\_Value ENDP

Main:

MOV AX, @DATA

MOV DS, AX

PUSH BP

MOV BP, SP ; save the position of SP for restore in the end

Lop:

MOV DX, OFFSET Menu

CALL Print\_String ;Print Menu string

CALL Get\_Num ; get the option from the user

; Switch ; compare the input from the user to 1-5 and jump to the currect oprion label

CMP AL, '5'

JE EndL

CMP AL, '4'

JE Maximum

CMP AL, '3'

JE Minimum

CMP AL, '2'

JE PopOp

; if we here, the option is 1

; Option 1: Push

CMP SP, 0

JE StackFull ; if the stack is full, jump to StackFull label

CALL Get\_Value ; get value from the user

CMP Buffer, 0

JA MaxMin ; if buffer is equal to zero that means the stack is empty

MOV BH, AL ; so we set max and min to be that value from the user

MOV BL, AL

PUSH AX ; push the value from the user to the stack

PUSH BX ; push our current max and min

INC Buffer

JMP Lop ; jump to the begining of the loop

MaxMin: ; Check if the current value we received is maximum or minimum in the stack

POP BX ; Get our Max and Min from stack and return it to stack

PUSH BX

MAX: ; compare the current value to the max value we have in stack and place it in BH if its higher

CMP AL, BH

JB MIN

MOV BH, AL

MIN: ; compare the current value to the min value we have in stack and place it in BL if its lower

CMP AL, BL

JA AfterFind

MOV BL, AL

AfterFind:

PUSH AX ; push the value from the user to stack

PUSH BX ; push our new current max and min to stack

INC Buffer

JMP Lop

PopOp: ; Option 2: POP

CMP Buffer, 0 ; if buffer is zero that means we dont have anything in stack,

JE StackEmp ; jump to StackEmp

POP AX ; pop the old max and min from stack

POP AX ; pop the latest value from the user to AX and print it

CALL Dos\_Print

DEC Buffer ; buffer--

JMP Lop

Minimum: ; Option 3: Min

CMP Buffer, 0 ; if buffer is equal to zero

JE StackEmp ; jump to StackEmp

POP BX ; Get our Max and Min from stack and return it to stack

PUSH BX

MOV AL, BL ; move the min value to AL and print it

CALL Dos\_Print

JMP Lop

Maximum: ; Option 4: Max

CMP Buffer, 0 ; if buffer is equal to zero

JE StackEmp ; jump to StackEmp

POP BX ; Get our Max and Min from stack and return it to stack

PUSH BX

MOV AL, BH ; move the max value to AL and print it

CALL Dos\_Print

JMP Lop

StackFull: ; StackFull label - prints StackF string and jump to the begining of the loop

MOV DX, OFFSET StackF

CALL Print\_String

JMP Lop

StackEmp: ; StackEmp label - prints StackE string and jump to the begining of the loop

MOV DX, OFFSET StackE

CALL Print\_String

JMP Lop

WrongInp: ; WrongInp label - prints WrongInput string and jump to the begining of the loop

MOV DX, OFFSET WrongInput

CALL Print\_String

JMP Lop

EndL: ; the end of the loop, we need to restore the address of the

MOV SP, BP ; set to the address in BP (one before the beginning of the stack)

POP BP ; get BP value back from stack and SP now is set to the begining of the stack

EndPro:

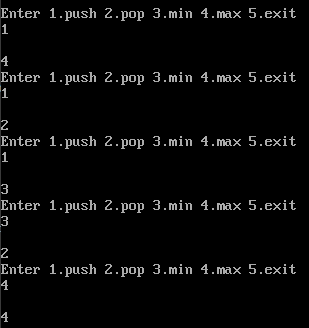
MOV AH,4Ch      ; Set terminate option for int 21h

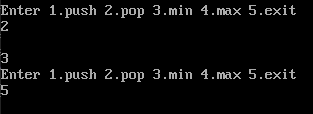
INT 21h         ; Return to DOS (terminate program)

END Main

**פלטים:**

**כמו בדוגמא:**





**קלט לא תקין:**

