EXPERIMENT 2

/\* Design, Develop and Implement a menu driven C program for the following operations on STACK of Integers

(Array Implementation of stack with maximum size MAX).

a. Push an element on to stack

b. Pop an element from stack

c. Demonstrate Overflow and Underflow situations on stack

d. Display the status of stack

e. Exit

Support the program with appropriate functions for each of the above operations. \*/

#include<stdio.h>

#include<stdlib.h>

#define MAX 5

int stack[MAX], top=-1;

void push(int);

void pop();

void display();

int main()

{

int choice=1,item;

while(choice)

{

printf("\n\n........STACK OPERATIONS.........\n");

printf("1. Push\n");

printf("2. Pop\n");

printf("3. Display\n");

printf("4. Exit\n");

printf(".....................................\n");

printf("Enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("Enter the element to be inserted : \t");

scanf("%d",&item);

push(item);

break;

case 2: pop();

break;

case 3: display();

break;

case 4:exit(0);

default: printf("\n Invalid choice\n");

}

}

return 0;

}

void push(int item)

{

if(top==(MAX-1))

{

printf("\nSTACK OVERFLOW\n");

return;

}

top=top+1;

stack[top]=item;

}

void pop()

{

int item;

if(top==-1)

{

printf("\nSTACK UNDERFLOW\n");

return;

}

item=stack[top];

top=top-1;

printf("The poped element is %d\t" , item);

}

void display()

{

int i;

if(top==-1)

{

printf("\n Stack is empty and nothing to display\n");

return;

}

printf("\n The stack elements are: \n");

for(i=top;i>=0;i--)

{

printf("%d\n",stack[i]);

}

}

/\* OUTPUT:

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 10

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 20

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 30

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 40

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 50

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

3

The stack elements are:

50

40

30

20

10

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

1

Enter the element to be inserted : 60

STACK OVERFLOW

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

The poped element is 50

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

The poped element is 40

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

The poped element is 30

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

The poped element is 20

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

3

The stack elements are:

10

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

The poped element is 10

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

2

STACK UNDERFLOW

........STACK OPERATIONS.........

1. Push

2. Pop

3. Display

4. Exit

.....................................

Enter your choice

4

Process returned 0 (0x0) execution time : 67.988 s

Press any key to continue.

\*/