IMPLEMENTATION OF STACK USING ARRAY

Name: Aditi Paretkar

Roll: 1911034

#include <stdio.h>

#define MAX 5

struct Stack

{

int top;

int a[MAX];

};

int push(struct Stack\*);

int pop(struct Stack\*);

int peek(struct Stack\*);

int display(struct Stack\*);

int main(void) {

struct Stack s;

struct Stack \*a=&s;

s.top=-1;

int c;

do {printf("\nEnter your choice\n");

printf("\n1 to push");

printf("\n2 to pop");

printf("\n 3 to peek");

printf("\n 4 to display");

printf("\n -1 to exit");

scanf("%d",&c);

switch (c)

{

case 1 :

{

push(a);

}

break;

case 2 :

{

pop(a);

}

break;

case 3 :

{

peek(a);

}

break;

case 4:

{

display(a);

}

}

}

while(c!=-1);

}

int push( struct Stack \*b)

{int val=0;

if(b->top==MAX-1)

{

printf("Stack is completely full");

}

else

{

printf("\n Enter the no. to be pushed on the stack");

scanf("%d",&val);

b->top=b->top+1;

b->a[b->top]=val;

//since we only pass the pointer to the struct , thus any object of the struct that we reference has to be incremented with the help of the pointer itself

}

}

int pop(struct Stack \*f)

{

if(f->top!=-1)

{

printf("The value popped off the stack will be %d",f->a[f->top]);

f->top--;

}

else

{

printf("Stack is already empty");

}

}

int peek(struct Stack \*d)

{

if(d->top!=-1)

{

printf("The element at the top is %d",d->a[d->top]);

}

else printf("The stack is empty");

}

int display(struct Stack \*e)

{int temp;

if(e->top!=-1)

{temp=e->top; //temp variable is used in order to avoid making changes in the top variable while displaying the stack , else the top variable will always be decremented just to display the stack

while(temp!=-1)

{

printf("\n%d",e->a[temp]);

temp--;

}

}

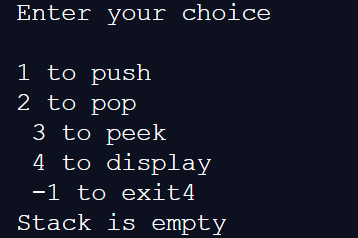
else

printf("Stack is empty");

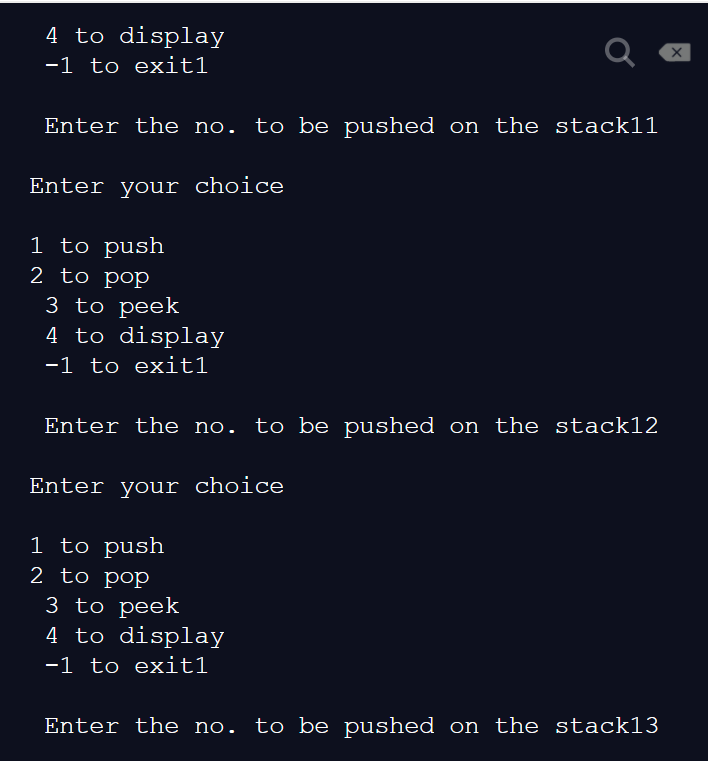
}

Screenshots of the output :

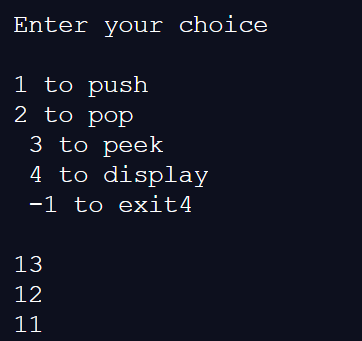
Case 1: Displaying the stack when it is empty (no numbers on the stack):



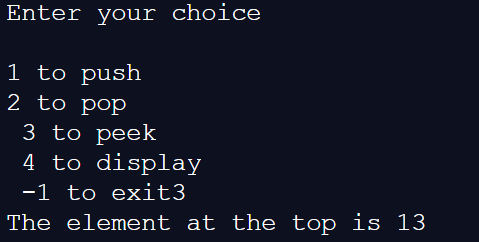
Case 2: pushing few numbers on the stack:



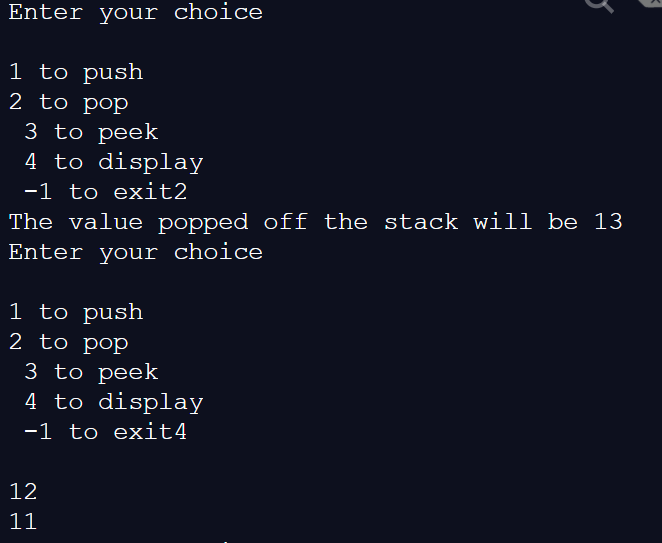
Case 3 : displaying the stack:



Case 4: peeking of stack (shows the topmost element):

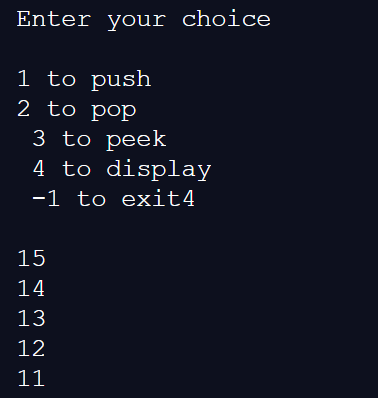


Case 5: popping of a number from the stack and displaying the stack after popping :

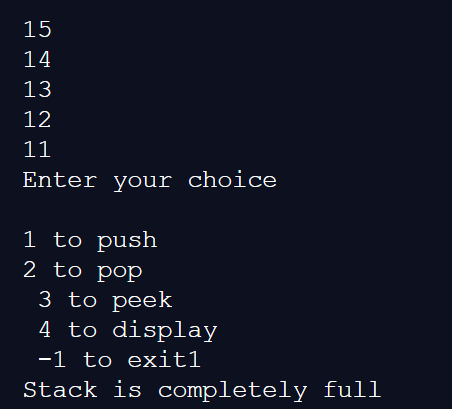


Case 6: after filling stack to MAX capacity(which is 5 in this case) , it will show that no more numbers can be entered :

For this , we first fill the stack to its max capacity and then display the stack :



Now we cannot enter any more numbers on the stack as it is already full:



Last case: popping off numbers from an empty stack :

