**#3. Stack**

**Roll Number: CB.EN.P2EBS22007**

**Date of Submission: 21-11-2022**

**Aim:**

To perform following Stack operations using C Programming:

1. Stack Creation
2. Push an element
3. Pop an element
4. Peek the stack
5. Check the status of the stack full/empty
6. View the entire stack after each of the above operation

**Tools Required:**

Text editor with C Compiler.

**Experiment:**

Code

#include <stdio.h>

#include <stdlib.h>

typedef struct stack1

{

int top;

unsigned int size;

int \*array;

}S1;

struct stack1 create(unsigned int n)

{

S1 \*sp1;

sp1=(struct stack1\*)malloc(sizeof (struct stack1));

sp1->array=(int\*)malloc(n\*sizeof (int));

sp1->top=-1;

sp1->size=n;

return \*sp1;

}

int isfull(S1 \*s1)

{

if(s1->top==s1->size-1){

printf("\n\n----------Cannot enter the element: Stack is full------------------\n\n");

return 0;

}

else{

printf("stack is not full\n");

return 1;

}

}

int isEmpty(S1 \*s1){

if(s1->top==-1){

printf("\n\n-------------------Stack is Empty----------------\n\n");

return 0;

}

else{

printf("\n| Stack is not empty |\n");

return 1;

}

}

void push(S1 \*s1)

{

if(isfull(s1))

{

int number,a;

printf("Enter the element:");

scanf("%d",&number);

s1->array[++s1->top]=number;

}

}

void pop(S1 \*s1){

if(isEmpty(s1)){

printf("%d removed at positon %d",s1->array[s1->top],s1->top);

s1->top=s1->top-1;

}

}

void peek(S1 \*s1){

int peekValue;

printf("Enter the stack element to be viewed:");

scanf("%d",&peekValue);

printf("%d",s1->array[peekValue]);

}

void stackView(S1 \*s1){

int i;

for (i=s1->size;i>=0;i--){

printf("\n%d\n", s1->array[i]);

}

}

int main()

{

S1 \*s1;

unsigned int n,userInput;

int i;

printf("\nEnter size of the stack:");

scanf("%d", &n);

\*s1=create(n);

while(1){

printf("\n======================OPTIONS=========================");

printf("\n1-Push\n2-Pop\n3-Peek\n4-isFull()\n5-isEmpty()\n6-View Stack\n7-Exit\n");

printf("Enter the option:");

scanf("%d",&userInput);

switch(userInput){

case 1:push(s1);

break;

case 2:pop(s1);

break;

case 3:peek(s1);

break;

case 4:isfull(s1);

break;

case 5:isEmpty(s1);

break;

case 6:

printf("-------STACK--------\n");

stackView(s1);

case 7: break;

default: printf("Enter a valid number");

}

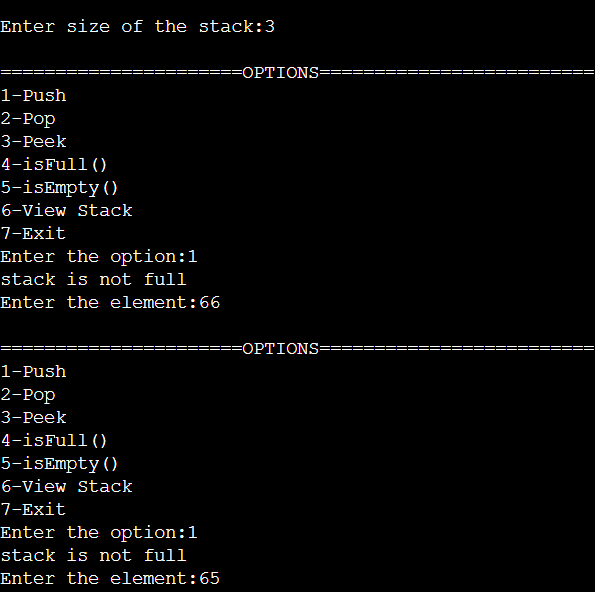
}

return 0;

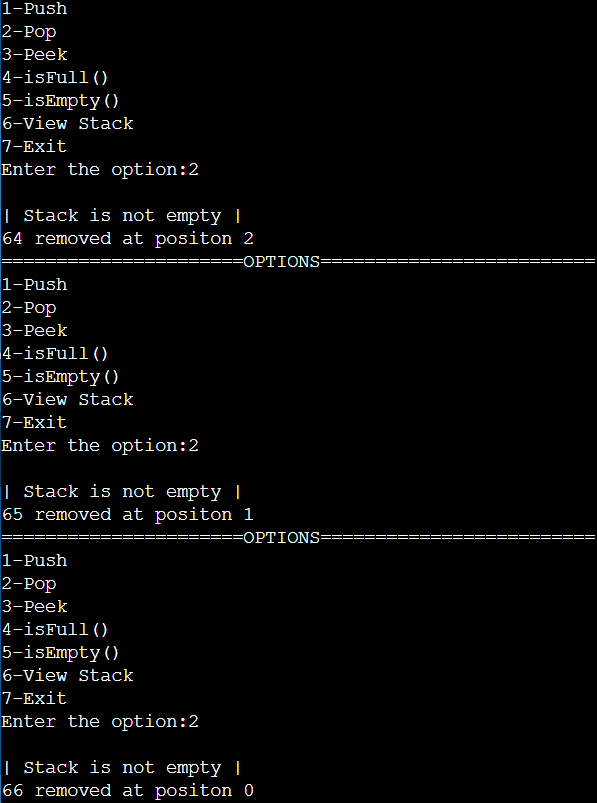
}

Result

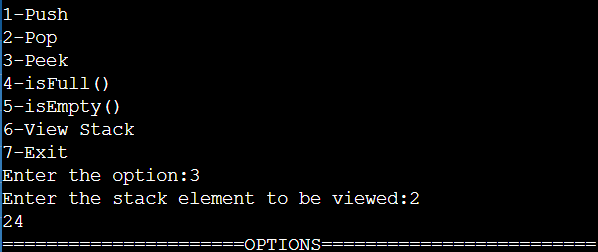
1. Push



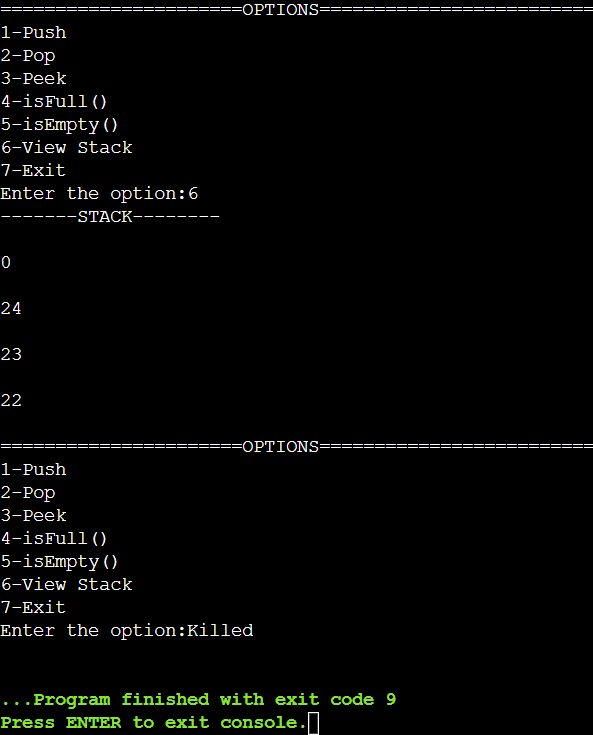
1. Pop



1. Peek



1. View Stack



**Inference and Result:**

Stack is implemented in C and stack operations are performed and result is observed.