**#3. Stack**

**Roll Number: cb.en.p2ebs22006**

**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;

}St1;

struct stack1 create(unsigned int n)

{

St1 \*Sta1;

Sta1=(struct stack1\*)malloc(sizeof (struct stack1));

Sta1->array=(int\*)malloc(n\*sizeof (int));

Sta1->top=-1;

Sta1->size=n;

return \*Sta1;

}

int isfull(St1 \*st1)

{

if(st1->top==st1->size-1){

printf("\n\n#######Stack is full#########\n\n");

return 0;

}

else{

printf("stack is not full, you can enter %d elements\n",(st1->size-1-st1->top));

return 1;

}

}

int isEmpty(St1 \*st1){

if(st1->top==-1){

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

return 0;

}

else{

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

return 1;

}

}

void push(St1 \*s1)

{

if(isfull(s1))

{

int number,a;

printf("Enter the element:");

scanf("%d",&number);

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

}

}

void pop(St1 \*s1){

if(isEmpty(s1)){

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

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

}

}

void peek(St1 \*s1){

int peekValue;

printf("Enter the stack element which needs to be viewed:\n");

scanf("%d",&peekValue);

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

}

int view(St1 \*st1)

{

unsigned int i;

for(i=0;i<st1->size;i++)

{

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

if(st1->array[i]==0)

{

printf(" | |\n");

}

else

{

printf(" |%d|\n",st1->array[i]);

}

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

}

}

int main()

{

St1 \*s1;

unsigned int n,userInput;

printf("Enter size of the stack:\n");

scanf("%d", &n);

\*s1=create(n);

while(1){

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

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: view(s1);

break;

case 7: exit(1);

break;

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

}

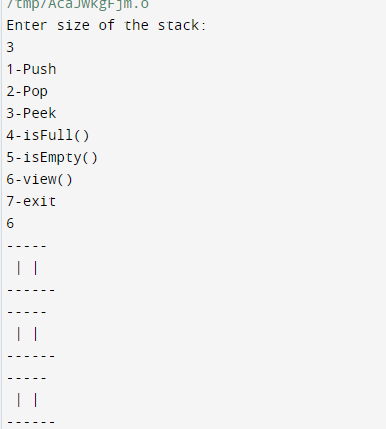
}

return 0;

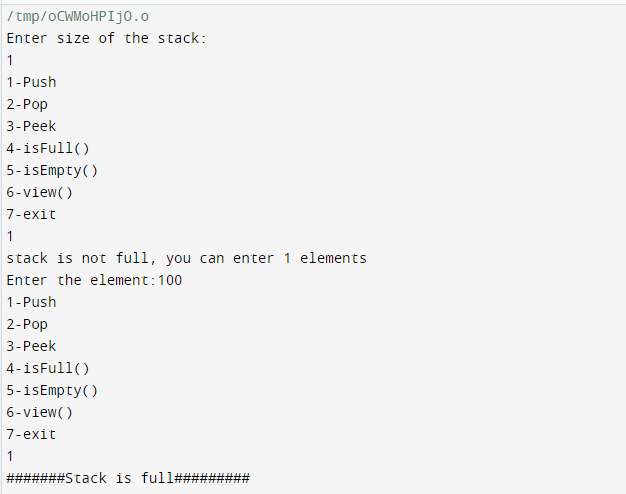
}

Result:

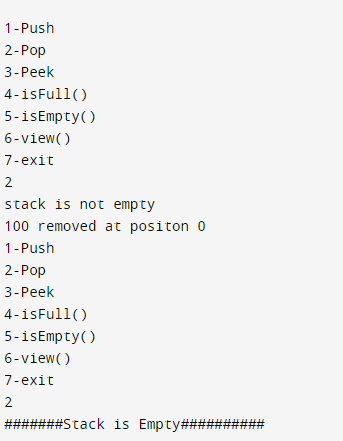
1. Stack Creation



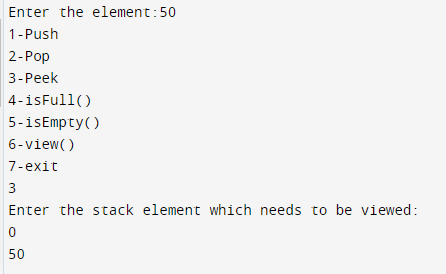
1. Push an element:



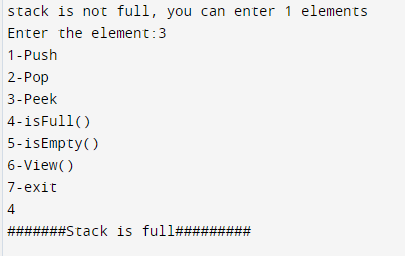
1. Pop an element



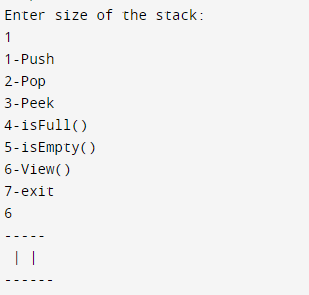
1. Peek the Stack

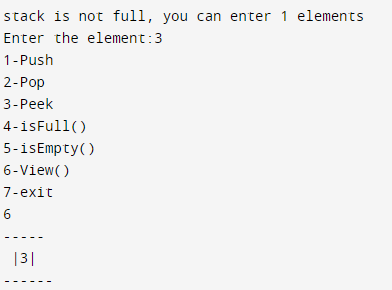


1. Check the status of the stack full/empty



1. View the entire stack after each of the above operation





**Inference and Result:**

Implementation of stack and the functions such as creation, push, pop, peek, view and checking whether the stack is full or not has been done and the output has been obtained and shown.