**#3. Stack**

**Roll Number:**

**Date of Submission:**

**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;

S1 create(unsigned int n)

{

S1 \*stm1;

stm1=(struct stack1\*)malloc(sizeof (struct stack1));

stm1->array=(int\*)malloc(n\*sizeof (int));

stm1->top=-1;

stm1->size=n;

return \*stm1;

}

int isfull(S1 \*s1)

{

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

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

return 0;

}

else{

printf("stack is not full, you can enter %d elements\n",(s1->size-1-s1->top));

return 1;

}

}

int isEmpty(S1 \*s1){

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

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

return 0;

}

else{

printf("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->array[s1->top]=NULL;

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

}

}

void peek(S1 \*s1){

int peekValue;

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

}

void view(S1 \*s1){

int i;

printf("%d",s1->size);

for(i=s1->size-1;i>=0;i--){

if(!s1->array[i]){

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

printf("|\t");

printf("\t");

printf("|\n");

printf("---------");

}

}

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

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

printf("|\t");

if(!s1->array[i])

printf("\t");

if(s1->array[i])

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

printf("|\n");

printf("---------");

}

}

int main()

{

S1 \*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);

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

}

}

return 0;

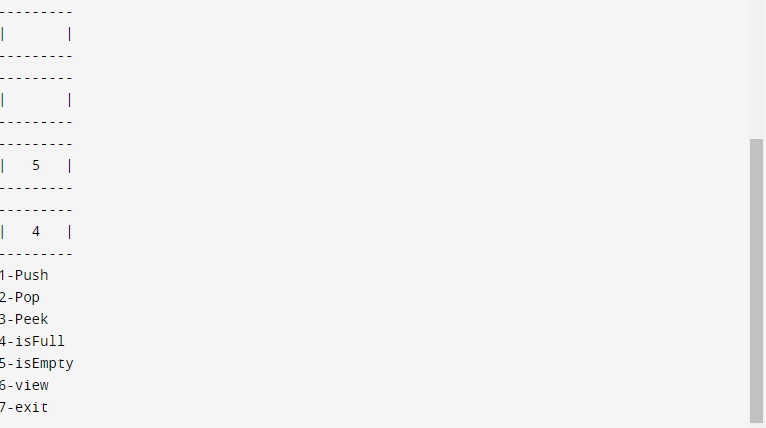
}

Result

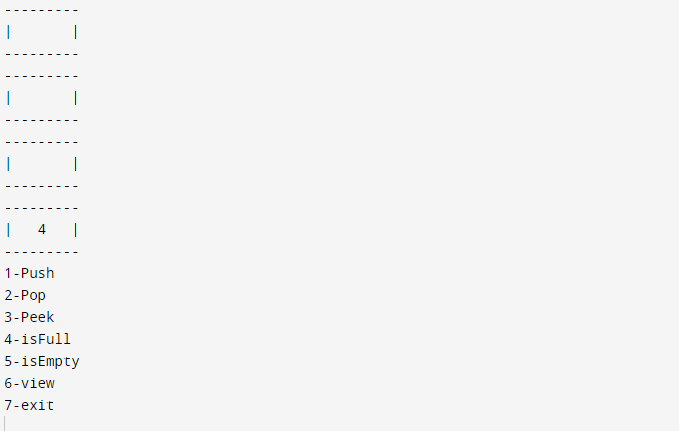
1. Create an empty stack



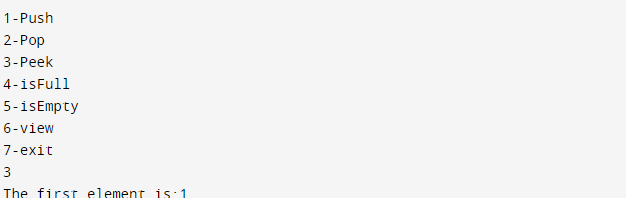
1. Pushed two elements in stack



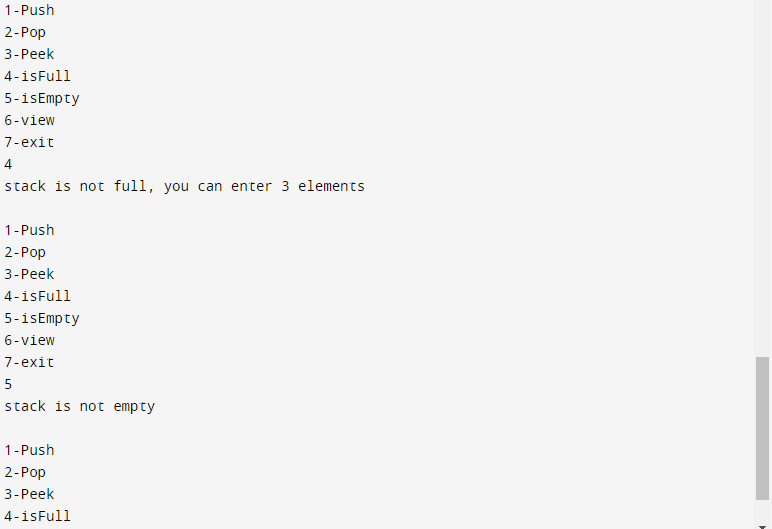
1. After pop:



1. Peek stack



1. View status



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



**Inference and Result:**

Thus, have implemented stack using arrays and implemented various functions such as push, pop, peek, view, isFull, isEmpty. Thus a concept on linear data structures has been learned.