Ex. No. 1a **Stack Array**

Date:

**Aim:**

To implement stack operations using array.

**Algorithm:**

1. Start

2. Define a array stack of size max = 5

3. Initialize top = -1

4. Display a menu listing stack operations

5. Accept choice

6. If choice = 1 then

If top < max -1

Increment top

Store element at current position of top

Else

Print Stack overflow

Else If choice = 2 then

If top < 0 then

Print Stack underflow

Else

Display current top element

Decrement top

Else If choice = 3 then

Display stack elements starting from top

7. Stop

**Program:**

/\* Stack Operation using Arrays \*/

#include <stdio.h>

#include <stdlib.h>

#define max 5

static int stack[max];

int top = -1;

void push(int x)

{

stack[++top] = x;

}

int pop()

{

return (stack[top--]);

}

void view()

{

int i;

if (top < 0)

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

else

{

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

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

{

printf("%4d", stack[i]);

}

printf("\n");

}

}

void main()

{

int ch=0, val;

system("clear");

while(ch != 4)

{

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

printf("1.PUSH ");

printf("2.POP ");

printf("3.VIEW ");

printf("4.QUIT \n");

printf("Enter Choice : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

if(top < max-1)

{

printf("\nEnter Stack element : ");

scanf("%d", &val);

push(val);

}

else

printf("\n Stack Overflow \n");

break;

case 2:

if(top < 0)

printf("\n Stack Underflow \n");

else

{

val = pop();

printf("\n Popped element is %d\n", val);

}

break;

case 3:

view();

break;

case 4:

exit(0);

default:

printf("\n Invalid Choice \n");

}

}

}

**Output:**

**Result**

Thus push and pop operations of a stack was demonstrated using arrays.