

01/01/2024

Prog - 01

Q.1. Write a programs the working of stacks using an array with the following

a) Push

b) Pop

c) Display

The program should print appropriate message for stack overflow, Stack Underflow

→

```
#include <stdio.h>
#include <conio.h>
```

```
int STK[MAX], top = -1;
```

→

```
#include <stdio.h>
```

```
#include <stdio.h>
```

```
#define N 5
```

```
int stack[N];
```

```
int top = -1;
```

```
Void push()
```

```
{
```

```
if (top == N - 1);
```

```
{
```

```
printf ("Stack overflow");
```

```
}
```

```
else
```

```
{
```

```
int x;
```

```
printf ("Enter the element to be inserted");
```

```
scanf ("%d", &x);
```

```
top++;
```

```
STK [top] = x;
```

```
3
```

```
3
```

01/01/2024

```
void pop()
{
    if (top == -1)
    {
        printf("Stack underflow");
    }
    else
    {
        int y;
        y = stack[top];
        top--;
        printf("The element deleted is %d", y);
    }
}
```

```
void display()
{
    if (top == -1)
    {
        printf("Stack is empty");
    }
    else
    {
        printf("The elements in Stack are : ");
        for (int i = N, i >= 0; i--)
        {
            printf("%d", stack[i]);
        }
    }
}
```

```
Void main()
{
    while (1)
```

6/10/2024

{

int choice;

printf ("Enter your choice = In

1. push in

2. pop in

3. Display ");

Scang ("I. d", &choice);

Switch (choice)

{

case 1 : push ();

break;

case 2 : pop ();

break;

case 3 : display ();

break;

default : printf ("Entered element is invalid");

break;

}

}

}

Output

Enter the operation 1. push

2. pop

3. Display

4. -1 to exit.

1. Enter the number

7

successfully pushed.

~~entered
11/10/2024~~

enter the operation

1.push

2.pop

3.display

enter -1 to stop

1

enter the values

10

push operation is succesfull

1

enter the values

20

push operation is succesfull

2

20 pop() operation successfull

3

10

-1

stopping the operations

Process returned 0 (0x0) execution time : 24.343 s

Press any key to continue.

01/01/2024

Prog-02.

- Q2. WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operation +, -, *, /.

→ #include <stdio.h>

#include <ctype.h>

#define size 50

char stack [size];

int top = -1;

push (char elem)

{

stack [++top] = elem;

}

char pop()

{

return (stack [top--]);

}

int pr (char symbol)

{

if (symbol == '+')

{

return (3);

}

else if (symbol == '*' || symbol == '/')

{

return (2);

}

else if (symbol == '-' || symbol == '/')

{

return (1);

}

01/01/2024

```
else
{
    return (0);
}

Void main()
{
    Char infix[50], postfix[50] , ch, elem;
    int i=0, k=0;
    printf ("Enter the infix expression:");
    Scanf ("%s", infix);
    push ('#');
    while (ch != infix[i++]) != '\0'
    {
        if (ch == 'c') push (ch);
        else
            if (isalnum(ch)) postfix[k++] = ch;
        else
            if (ch == ')')
            {
                while (stack [top] != '(')
                    postfix [k++] = pop();
                elem = pop();
            }
        else
            {
                while (pr (stack [top]) >= pr (ch))
                    postfix [k++] = pop ();
                push (ch);
            }
    }
}
```

01/01/2024

Page 1 of 1

while (stack [top] != '#')

postfix [k++] = pop();

postfix [k] = '10';
printf ("In postfix expression = %s\n", postfix);

Output

(K+L-M*N+(O^P)*W/U/V*T+Q)

Infix Expression:

(K+L-M*N+(O^P)*W/U/V*T+Q)

Postfix Expression:

KL+MN*-OP^W^U/V/T^+Q+

Enter size of stack 3

Assume the infix expression contains single letter variables and single digit constants only.

Enter Infix expression : k*l+m(n)

Postfix Expression: kl*mn+

Process returned 0 (0x0) execution time : 49.048 s

Press any key to continue.