

PROGRAM - 1

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

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

```
#define SIZE 100
```

```
void push(int);
```

```
void pop();
```

```
void display();
```

```
int stack[SIZE], top = -1;
```

```
void push(int value) {
```

```
    if (top == SIZE - 1)
```

```
        printf("\n Stack is Full");
```

```
    else {
```

```
        top = top + 1;
```

```
        stack[top] = value;
```

```
    }
```

```
}
```

```
void pop() {
```

```
    if (top == -1) {
```

```
        printf("\n stack is empty");
```

```
    else {
```

```
        printf("\n Deleted : %d", stack[top]);
```

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

```
}
```

```
}
```

Handwritten notes:
Printed
11/2/2024


```
void display () {
```

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

```
    Print ("\n stack is empty!");
```

```
else {
```

```
    int i;
```

```
    printf ("\n stack elements are: \n");
```

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

```
        printf ("%d \n", stack[i]);
```

```
}
```

```
}
```

```
void main () {
```

```
int value, choice;
```

```
while (1) {
```

```
    printf ("\n\n *** MENU *** \n");
```

```
    printf ("1. Push \n, 2. Pop \n 3. Display \n 4. Exit");
```

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

```
    switch (choice) {
```

```
case 1: printf ("Enter the value to be inserted:");
```

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

```
        push (value);
```

```
        break;
```

```
case 2: pop ();
```

```
        break;
```

```
case 3: display ();
```

```
        break;
```

```
case 4: exit (0);
```

```
default: printf ("\n ERROR \n");
```

```
}
```

```
}
```

```
}
```


OUTPUT:

*** MENU ***

1. PUSH

2. POP

3. Display

4. Exit

1

Enter value to be inserted : 2

** MENU **

1. PUSH

2. POP

3. DISPLAY

4. Exit

2

Enter value to be inserted : 2

** MENU **

:

:

3

Stack Elements are: 1 2

2

1

** MENU **

:

:

2

Deleted : 2

** MENU **

:

4

Program 2: (Infix to postfix)

```
#include <stdio.h>
#include <ctype.h>
#define SIZE 30
char Stack[SIZE];
int top = -1;

void Push (char ele)
{
    stack[++top] = ele;
}

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);
    else
        return (0);
}

void main () {
    char infix[50], postfix[50], ch, ele;
    int i = 0, k = 0;
    printf("Enter 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] != 'c') {

Postfix[k++] = pop(); }

ch = pop();

}

else {

while (pr(stack[top]) >= pr(ch)) {

Postfix[k++] = pop(); }

Push(ch);

}

}

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

Postfix[k++] = pop(); }

Postfix[k] = '\0';

Printf("The postfix expression is: %s", postfix);

}

Output

Enter infix expression: a+b+c

The postfix expression is: abc++