NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: F
ROLL NO.: 230701376

PROGRAM: Infix to postfix conversion

Write a C program to perform infix to postfix conversion using stack.

```
#include<stdio.h>

#include<string.h>


#define size 10


int top=-1;

char input[size];

char post[size]="";


int of()

{

   if (top==size-1)

      return 1;

   else

      return 0;

}


int uf()

{

   if (top==-1)
```

```
        return 1;
    else
        return 0;
}

void push(char rol)
{
    if (of())
    {
        printf("Overflow");
    }
    else
    {
        top=top+1;
        input[top]=rol;
    }
}

char pop()
{
    if (uf())
    {
        printf("Underflow");
```

```
    }
    else{
        char del=input[top];
        top=top-1;
        return del;
    }
}

int opd(char rol)
{
    switch(rol)
    {
        case '+':
        case '-':
            return 2;

        case '*':
        case '/':
            return 3;

        case '^':
            return 4;
```

```
        case '(':
            return 1;
    }
    return 0;
}

int ipd(char rol)
{
    switch (rol)
    {
        case '+':
        case '-':
            return 1;

        case '*':
        case '/':
            return 2;

        case '^':
            return 3;

        case '(':
            return 4;
```

```
    }

    return 0;

}


int main()

{

    char temp;

    printf("Enter the infix expression: ");

    scanf("%s",input);


    for (int i=0; input[i]!='\0'; i++)

    {

        if (input[i]>='a' && input[i]<='z')

            printf("%c", input[i]);

        else if (input[i] == '(')

            push(input[i]);

        else if (input[i] == ')')

        {

            while (input[top] != '(')

            {

                temp = pop();

                printf("%c", temp);

            }
```

```
            temp = pop();
        }
        else
        {
            while (!uf() && ipd(input[i]) <= opd(input[top]) && input[top] != '(')
            {
                temp = pop();
                printf("%c", temp);
            }
            push(input[i]);
        }
    }



    while (!uf())
    {
        temp = pop();
        printf("%c", temp);
    }


    printf("\n");
    return 0;
}
```