Week2: Write a program to convert infix to postfix using stacks.

```
#include<stdio.h>
#include<string.h>
int F(char symbol)
{
    switch(symbol)
    {
        case '+':
        case '-':
            return 2;

        case '*':
        case '/':
            return 4;

        case '^':
        case '$':
            return 5;

        case '(':
            return 0;

        case '#'
            return -1;

        default:
            return 8
    }

}
```

Date ___/___/_____

Saathi

```
int in(char symbol)
{
    switch(symbol)
    {

      case '+':
      case '-':
        return 1;

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

      case '^':
      case '$':
        return 6;

      case 'c':
        return 9;

      case '(':
        return 0

      default
        return 7

    }

}
```

Page No. ____

Saathi

Date / /

```
void infix_postfix(char infix[], char postfix[])
{
    int top, i, j;
    char s[30], Symbol;
    top = -1;
    s[++top] = '#';
    j = 0;
    for(i=0; i< strlen(infix); i++)
    {
        Symbol = infix[i];
        while(F[s[top]] > G[Symbol])
        {
            postfix[j] = s[top--];
            j++;
        }
        if(F[s[top]] != G[Symbol])
            s[++top] = Symbol;
        else
            top--
    }
    while(s[top] != '#')
    {
        postfix[j++] = s[top--];
    }

void main()
{
    char infix[20];
    char postfix[20];
```

Date __/__/____

Saathi

(7)
```
printf("Enter the valid expression\n");
scanf("%s", infix);
infix_postfix(infix, postfix);
printf("postfix is \n");
printf("%s\n", postfix);
}
```

OUTPUT

Enter the valid Expression

(A + (B-C) * D)

Post fix Expression is

ABC - D* + .