WEEK 2- Infix to postfix

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

char stack[100];
int top = -1;

void push(char y)
{
    stack[++top] = y;
}

char pop()
{
    if(top == -1)
        return -1;
    else
        return stack[top--];
}

int precedence(char y)
{
    if(y == '(')
        return 0;
    if(y == '+' || y == '-')
        return 1;
    if(y == '*' || y == '/')
        return 2;
    return 0;
}
```

```c
int main()
{
    char exp[100];
    char *ex, y;
    printf("Enter infix expression : ");
    scanf("%s",exp);
    printf("postfix expression is:\n");
    printf("\n");
    ex = exp;

    while(*ex != '\0')
    {
        if(isalnum(*ex))
            printf("%c ",*ex);
        else if(*ex == '(')
            push(*ex);
        else if(*ex == ')')
        {
            while((y = pop()) != '(')
                printf("%c ", y);
        }
        else
        {
            while(precedence(stack[top]) >= precedence(*ex))
                printf("%c ",pop());
            push(*ex);
        }
        ex++;
    }
```

```c
    while(top != -1)

    {

        printf("%c ",pop());

    }return 0;
```

```
Enter infix expression : ((a+b)*(c-d))/e
postfix expression is:

a b + c d - * e /


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
}
```