

## 12a.expression conversion :-

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

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

```
char stack[100];
```

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

```
void push(char x)
```

```
{
```

```
    stack[++top] = x;
```

```
}
```

```
char pop()
```

```
{
```

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

```
        return -1;
```

```
    else
```

```
        return stack[top--];
```

```
}
```

```
int priority(char x)
```

```
{
```

```
    if(x == '(')
```

```
        return 0;
```

```
    if(x == '+' || x == '-')
```

```
        return 1;
```

```
    if(x == '*' || x == '/')
```

```
        return 2;
```

```
    return 0;
```

```
}
```

```

int main()
{
    char exp[100];
    char *e, x;
    printf("Enter the expression : ");
    scanf("%s",exp);
    printf("\n");
    e = exp;

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

    while(top != -1)
    {

```

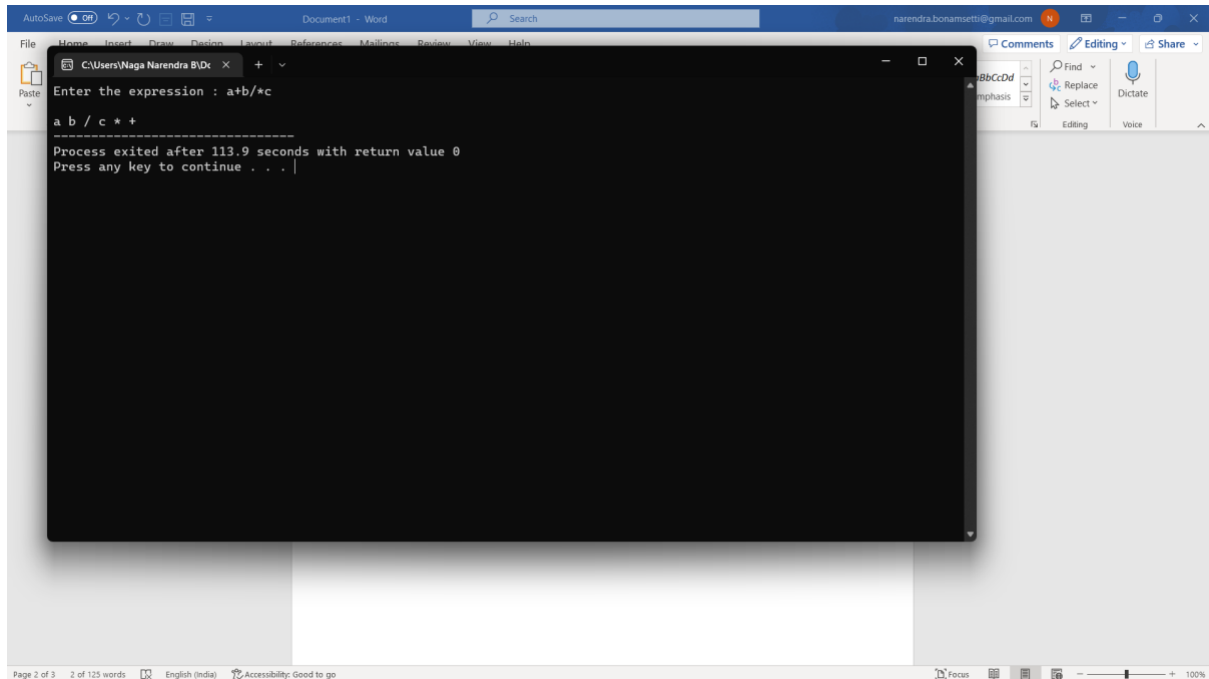
```

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

}return 0;

}

```



## 12b.expression evaluation :-

```

#include<stdio.h>

int top = -1, stack [100];

main ( ){

    char a[50], ch;

    int i,op1,op2,res,x;

    void push (int);

    int pop( );

    int eval (char, int, int);

    printf("enter a postfix expression:");

    gets (a);

    for(i=0; a[i]!='\0'; i++){

```

```

    ch = a[i];
    if (ch>='0' && ch<='9')
        push('0');
    else{
        op2 = pop ( );
        op1 = pop ( );
        res = eval (ch, op1, op2);
        push (res);
    }
}
x = pop ( );
printf("evaluated value = %d", x);
int ( );
}

void push (int n){
    top++;
    stack [top] = n;
}

int pop ( ){
    int res ;
    res = stack [top];
    top--;
    return res;
}

int eval (char ch, int op1, int op2){
    switch (ch){
        case '+' : return (op1+op2);
        case '-' : return (op1-op2);
        case '*' : return (op1*op2);

```

```

        case '/' : return (op1/op2);
    }
}

#include<stdio.h>

int top = -1, stack [100];

main ( ){

    char a[50], ch;

    int i,op1,op2,res,x;

    void push (int);

    int pop( );

    int eval (char, int, int);

    printf("enter a postfix expression:");

    gets (a);

    for(i=0; a[i]!='\0'; i++){

        ch = a[i];

        if (ch>='0' && ch<='9')

            push('0');

        else{

            op2 = pop ( );

            op1 = pop ( );

            res = eval (ch, op1, op2);

            push (res);

        }

    }

    x = pop ( );

    printf("evaluated value = %d", x);

    int ( );

}

void push (int n){

```

```

    top++;
    stack [top] = n;
}

int pop ( ){
    int res ;
    res = stack [top];
    top--;
    return res;
}

int eval (char ch, int op1, int op2){
    switch (ch){
        case '+' : return (op1+op2);
        case '-' : return (op1-op2);
        case '*' : return (op1*op2);
        case '/' : return (op1/op2);
    }
}

#include<stdio.h>

int top = -1, stack [100];

main ( ){
    char a[50], ch;
    int i,op1,op2,res,x;
    void push (int);
    int pop( );
    int eval (char, int, int);
    printf("enter a postfix expression:");
    gets (a);
    for(i=0; a[i]!='\0'; i++){
        ch = a[i];

```

```

    if (ch>='0' && ch<='9')
        push('0');
    else{
        op2 = pop ( );
        op1 = pop ( );
        res = eval (ch, op1, op2);
        push (res);
    }
}

x = pop ( );
printf("evaluated value = %d", x);
int ( );
}

void push (int n){
    top++;
    stack [top] = n;
}

int pop ( ){
    int res ;
    res = stack [top];
    top--;
    return res;
}

int eval (char ch, int op1, int op2){
    switch (ch){
        case '+' : return (op1+op2);
        case '-' : return (op1-op2);
        case '*' : return (op1*op2);
        case '/' : return (op1/op2);
    }
}

```

}

}

