

- 1) Write program to convert a given valid parenthesis sized infix arithmetic expression to postfix expression. The expression consists of single character operands & binary operators + (plus), - (minus), * (multiply), / (divide) & ^ (power).

```

→ #include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>
#define MAX 100
char st[MAX];
int top = -1;
void push(char st[], char);
char pop(char st[]);
void InfixtoPostfix(char source[], char Target[]);
int getPriority(char);
int main() {
    char infix[100], postfix[100];
    clrscr();
    printf("\n Enter any infix expression:");
    gets(infix);
    strcpy(postfix, "");
    InfixtoPostfix(infix, postfix);
    printf("\n The corresponding postfix expression is :");
    puts(postfix);
    getch();
    return 0;
}

void InfixtoPostfix(char source[], char Target
    [ ]) {

```

```

int i=0, j=0;
char temp;
strcpy(target, "");
while (source[i] != '\0') {
    if (source[i] == '(') {
        push(st, source[i]);
        i++;
    }
    else if (source[i] == ')') {
        while ((top != -1) && (st[top] != '('))
            target[j] = pop(st);
            j++;
    }
    if (top == -1) {
        printf("\n Invalid Expression");
        exit(1);
    }
    temp = pop(st);
    i++;
}
else if (isdigit(source[i]) || isalpha(source[i]) || source[i] == '-' || source[i] == '+') {
    target[j] = source[i];
    j++;
    i++;
}
else if (source[i] == '+' || source[i] == '-' || source[i] == '*' || source[i] == '/' || source[i] == '%') {
    while ((top != -1) && (st[top] != '(') && (getPriority(st[top]) > getPriority(source[i])))
        target[j] = pop(st);
        j++;
}

```



```

push(st, source[i]);
i++;

```

```

}

```

```

else {

```

```

    printf("\n Indirect element in Expression\n");

```

```

    exit(1);

```

```

}

```

```

}

```

```

while ((top != -1) && (st[top] != '(')) {

```

```

    target[j] = pop(st);

```

```

    j++;

```

```

}

```

```

target[j] = '\0';

```

```

}

```

```

int getPriority(char op) {

```

```

    if (op == '/' || op == '*' || op == '%')

```

```

        return 1;

```

```

    else if (op == '+' || op == '-')

```

```

        return 0;

```

```

}

```

```

void push(char st[], char val) {

```

```

    if (top == MAX - 1)

```

```

        printf("\n Stack Overflow");

```

```

    else {

```

```

        top++;

```

```

        st[top] = val;

```

```

    }

```

```

}

```

```

char pop(char st[]) {

```

```

    char val = ' ';

```

```

    if (top == -1)

```

```

        printf("\n Stack Underflow");

```

Date / /

Page

```
else {  
    val = st[top];  
    top--;  
}  
return val;  
}
```

Q18. Enter any infix expression : $A+B-C+D$
The corresponding postfix expression is : $AB+C-D$