```
#include <stdio.h>
#define N 100

int stack[N];

int top = -1;

void push(int item){
   if(top==N-1)
      printf("Stack overflow!\n");
   else
      stack[++top] = item;
}

int pop(){
   if(top==-1)
      printf("Stack underflow!\n");
   else
      return stack[top--];

}

int priority(char op){
   switch(op){
      case '*':   return 2;
                  break;
      case '/':   return 2;
                  break;
      case '+':   return 1;
```

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

int main()
{
    char s[50];
    char t[50];
    int l;
    int choice = 1;

    do{
        l = 0;
        printf("Enter your infix expression: ");
        scanf("%s", s);

        for(int i=0; s[i]!='\0'; i++){
            switch(s[i]){
                case '(':   push('(');
                        break;
                case ')':   while(stack[top]!='('){
                            t[l++] = pop();
                        }
                        pop();
                        break;
```

```
            case '*':
            case '/':
            case '+':
            case '-':   while(top!=-1 && priority(stack[top])>=priority(s[i])){
                           t[l++] = pop();
                        }
                        push(s[i]);
                        break;
            default: t[l++] = s[i];
          }
        }

        while(top!=-1){
           t[l++] = pop();
        }

        t[l] = '\0';

        printf("Postfix expression for \"%s\" is \"%s\".\n", s, t);

        printf("\n     :: Menu ::     \n1. Try another infix expression.\n2. Exit\nEnter your choice: ");

        scanf("%d", &choice);
    }while(choice!=2);

    return 0;
}
```

OUTPUT:

```
Enter your infix expression: vv
Postfix expression for "vv" is "vv".

     :: Menu ::
1. Try another infix expression.
2. Exit
Enter your choice: 1
Enter your infix expression: A+B
Postfix expression for "A+B" is "AB+".

     :: Menu ::
1. Try another infix expression.
2. Exit
Enter your choice: 1
Enter your infix expression: A+(B-c)
Postfix expression for "A+(B-c)" is "ABc-+".

     :: Menu ::
1. Try another infix expression.
2. Exit
Enter your choice: 1
Enter your infix expression: (A+(B-C)*D)
Postfix expression for "(A+(B-C)*D)" is "ABC-D*+".

     :: Menu ::
1. Try another infix expression.
2. Exit
Enter your choice: 2
```