NAME :- Aindrail Santra

Roll No :- 2029044

## ASSIGNMENT 6

3. WAP to convert an infix expression into its equivalent postfix notation

```cpp
#include<iostream>
using namespace std;
int prec( char c){
if(c=='^')
return 3;
else if(c=='*' || c=='/')
return 2;
else if(c=='-'||c=='+')
return 1;
else
return -1;
}
string infixToPostfix(string s){
stack st;
string res;
for( int i = 0 ; i if((s[i]>='a' && s[i]<='z') ||
(s[i]>='A' && s[i]<='Z')){
res+=s[i];
}
else if(s[i]=='('){
st.push(s[i]);
}
```

```cpp
else if(s[i]==')'){
while(! st.empty() &&  st.top()!='('){
res+=st.top();
st.pop();
}
if(! st.empty()){
st.pop();
}}
else{
while(! st.empty() &&  prec(st.top())>=prec(s[i])){
res+=st.top();
st.pop();
}
st.push(s[i]);
}}
while(! st.empty()){
res+=st.top();
st.pop();
}
return res;
}

int main(){
cout<<infixToPostfix("(a-b/c)*(a/k-l)");
```

```cpp
return 0;
}
```

OUTPUT
abc/-ak/1-*

4. WAP to convert an infix expression into its equivalent prefix notation.

```cpp
#include<iostream>
using namespace std;
int prec( char c){
if(c=='^')
return 3;
else if(c=='*' || c=='/')
return 2;
else if(c=='-'||c=='+')
return 1;
else
return -1;
}

string infixToPrefix(string s){
reverse(s.begin(), s.end());
stack st;
string res;
```

NAME :- Aindrail Santra

Roll No :- 2029044

```
for( int i = 0 ; i<s.length();i++){ if((s[i]>='a'&&
s[i]<='z') || (s[i]>='A' && s[i]<='Z')){
rest=s[i];
}
else if(s[i]==')'){
st.push(s[i]);
}
else if(s[i]=='('){
while(!st.empty() && st.top()!=')'){
rest=st.top();
st.pop();
}
if(!st.empty()){
st.pop();
}}
else{
while(!st.empty() && prec(st.top())>=prec(s[i])){
rest=st.top();
st.pop();
}
st.push(s[i]);
}
```

```cpp
}
while(!st.empty()){
res+ = st.top();
st.pop();
}
reverse(res.begin(), res.end());
return res;
}
int main(){
cout<<infixToPrefix("(a-b/c)*(a/k-l)")<<endl;
 return 0;
}
```

OUTPUT
*-a/bc-/akl