***INFIX TO POSTFIX :***

```
#include<bits/stdc++.h>
using namespace std;
string p,q;
stack<char>s1;
map<char,int>mp;
vector<int>v1,v2;
void InfixtoPostfix(string s)
{
   s1.push('(');
   s.push_back(')');
   for(int i=0;i<s.size();i++)
   {
      if(s[i]=='p'||s[i]=='q')
         q.push_back(s[i]);
      else if(s[i]=='^'||s[i]=='v'||s[i]=='~')
      {
         while(!s1.empty()&&s1.top()!='('&&s1.top()!=')'&&mp[s[i]]>mp[s1.top()])
         {
            char ch=s1.top();
            s1.pop();
            q.push_back(ch);
         }
         s1.push(s[i]);

      }
      else if(s[i]==')')
      {
         while(!s1.empty()&&s1.top()!='(')
         {
            char ch=s1.top();
            s1.pop();
            q.push_back(ch);
         }
         s1.pop();
      }
      else
         s1.push(s[i]);
   }

   while(!s1.empty())
      s1.pop();
}
```

```
int operation2(int a,int b,string x)
{
   stack<int>check;
   for(int i=0;i<p.size();i++)
   {
      if(x[i]=='p')
         check.push(a);
      else if(x[i]=='q')
         check.push(b);
      else if(x[i]=='~')
      {
         int c=check.top();
         c=!c;
         check.pop();
         check.push(c);
      }
      else if(x[i]=='v')
      {
         int d=check.top();
         check.pop();
         int c=check.top();
         check.pop();
         int ans=c|d;
         check.push(ans);
      }
      else if(x[i]=='^')
      {
         int d=check.top();
         check.pop();
         int c=check.top();
         check.pop();
         int ans=c&d;
         check.push(ans);
      }
   }
   return check.top();
}
```