Himanshu Dixit
2110326
B11

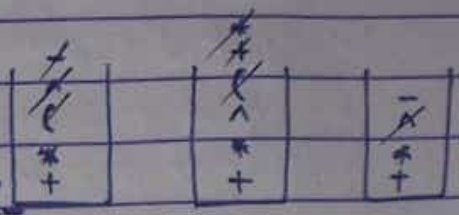
DS Assignment - 1

Q.1: a + b * (c ^ d - e) ^ (f + g * h) - i

result : a

ab

~~ab cd^e-fg*h+^*+i-~~



infix to postfix :

1. move left write
2. if character is found simply concat to result.
3. if symbol is found
   1. '(' push in stack
   2. ')' pop & add symbol to result
   3. check precedence with top symbol if precedence is less simply add to result and pop ( ) & so on at last after check with all symbol push to stack.
4. check whether symbol left in stack or not simply add to result.

```
#include <iostream>
#include <string.h>
using namespace std;

int precedence(char ch) {
if (ch == '^')
    return 3;
else if (ch == '*' || ch == '/')
    return 2;
else if (ch == '+' || ch == '-')
    return 1;
else return -1; }
```

```
void conversion ( string &s ) {

stack <char> &s ;
string result;
for(int i=0 ; i< s.length() ; i++)
 {
  char ch = s[i];
  if (ch >= 'a' && ch <= 'z' || ch >='A' && ch <='z')
   {
     result += ch ;
   }
  else if ( ch == '(' )
   s.push ( ch ) ;
  else if (ch == ')')
  {
   while ( s.top() != '(')
    {
      result += s.top() ;
      s.pop() ;
    }
   s.pop() ;
  }
  else {
  while ( !s.empty() && precedence (ch) <=
            precendence (s.top()) )
  {
     result += s.top() ;
     s.pop() ;
  }
  S.push (ch) ;
 }
}
```

```
while ( !st.empty () ) {
    result += s.top();
    s.pop();
}
cout << result ;
}

int main () {
string s;
cin >> s;
Conversion (s);
return 0;
}
```

Q.20

```
Node * Solution ( Node * head , int K ) {
  Node *newnode = new Node(k);
  if ( head == NULL ) {
    head = newnode ;
  }

  else if ( head -> data > K ) {
    newnode -> next = head ;
    head -> prev = newnode ;
    head = newnode ;
  }

  else {
  while ( curr -> next != NULL && curr -> next -> data < K )
  {
    curr = curr -> next ;
  }
  newnode -> next = curr -> next ;
  newnode -> prev = curr
  Node * n = curr -> next ;
  curr -> next = newnode ;
  if (n != NULL)
  {
    n -> prev = newnode ;
  }
}
return head ;
}
```

Q.5.

```
int * check ( int *arr , int n , int K )
{ int ans[2] = {-1, -1} ;
  int l=0 , r = n-1 ;
  int mid = (l+r)/2 ;
  while ( l<=r) {
    if ( arr[mid] == k ) {
    for ( int i = mid ; arr[i] == K && i>=0 ; i--)
     {
        ans[0] = i ;
     }
    for( int i = mid ; arr[i]==K && i<=n-1 ; i++ )
     {
        ans [i] = i ;
     }
  }
   else if ( arr[mid] < K )
    {
      l = mid+1 ;
    }
    else
    {
      r = mid-1 ;
    }
  }
  return ans ;
}
```