**Basic operation**

#include <bits/stdc++.h>

using namespace std;

struct node

{

    int data;

    struct node \*next;

};

struct node \*top=NULL;

struct node \*push(struct node \*top, int val)

{

    struct node \*ptr;

    ptr=(struct node\*)malloc(sizeof(struct node));

    ptr->data=val;

    if(top==NULL)

    {

        ptr->next=NULL;

        top=ptr;

    }

    else

    {

        ptr->next=top;

        top=ptr;

    }

    return top;

}

struct node \*display(struct node \*top)

{

    struct node \*ptr;

    ptr=top;

    if(top==NULL)

    {

        cout<<"stack is empty"<<endl;

    }

    else{

    while(ptr!=NULL)

    {

        cout<<ptr->data<<endl;

        ptr=ptr->next;

    }

    }

    return top;

}

struct node \*pop(struct node \*top)

{

    struct node \*ptr;

    ptr=top;

    if(top==NULL)

    {

        cout<<"stack is empty"<<endl;

    }

    else

    {

        top=top->next;

        cout<<"value to be deleted is "<<ptr->data<<endl;

    }

    return top;

}

void \*peek(struct node \*top)

{

    if(top==NULL)

    {

        cout<<"stack is empty"<<endl;

    }

    else{

        cout<<"top value is "<<top->data<<endl;

    }

}

int main()

{

    int option,val;

    do

    {

    cout<<"1 to push, 2 to display, 3 to pop, 4 to peek and 5 to exit"<<endl;

    cin>>option;

    switch(option)

    {

        case 1:

            cout<<"Enter a value"<<endl;

            cin>>val;

            top=push(top,val);

            break;

        case 2:

            top=display(top);

            break;

        case 3:

            top=pop(top);

            break;

        case 4:

            peek(top);

            break;

    }

    }while(option !=5);

    return 0;

}

**INFIX TO POSTFIX**

#include<bits/stdc++.h>

using namespace std;

int prec(char c)

{

    if(c=='^')

    return 5;

    else if(c=='/' || c=='\*' || c=='%')

    return 4;

    else if(c=='+'|| c=='-')

    return 3;

    else

    return 2;

}

void intopo(string s)

{

    int i;

    string result;

    stack<char> st;

    for(i=0;i<s.length();i++)

    {

        char c=s[i];

        if((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9'))

        {

            result +=c;

        }

        else if(c=='(')

        {

            st.push('(');

        }

        else if(c==')')

        {

            while(st.top()!='(')

            {

                result +=st.top();

                st.pop();

            }

            st.pop();

        }

        else

        {

            if(!st.empty() && prec(s[i])<=prec(st.top()))

            {

                result +=st.top();

                st.pop();

            }

            st.push(s[i]);

        }

    }

    while(!st.empty())

    {

        result +=st.top();

        st.pop();

    }

    cout<<result;

}

int main()

{

    string sl;

    sl="a+b\*(c^d-e)^(f+g\*h)-i";

    intopo(sl);

    return 0;

}

**INFIX TO PREFIX**

#include<bits/stdc++.h>

using namespace std;

int pre(char ch)

{

    if(ch=='+' || ch=='-')

    {

        return 1;

    }

    else if(ch=='/' || ch=='\*')

    {

        return 2;

    }

    elseif(ch=='^' || ch=='%')

    {

        return 3;

    }

    else

    {

        return 0;

    }

}

string infix\_to\_postfix(string infix)

{

    infix='('+infix+')';

    int i;

    stack<char> st;

    char ch;

    string output;

    for(i=0;i<infix.length();i++)

    {

        ch=infix[i];

        if(isalpha(ch) || isdigit(ch))

        {

            output +=ch;

        }

        else if(ch=='^' && prec(ch)<=)

    }

}

**NEXT GREATER RIGHT**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st;

    int n,i;

    cout<<"enter total values"<<endl;

    cin>>n;

    int ar[n],output[n];

    memset(output,n,0);

    cout<<"enter the values"<<endl;

    for ( i = 0; i < n; i++)

    {

        cin>>ar[i];

    }

    for(i=n-1;i>=0;i--)

    {

        if(st.empty())

        {

            output[i]=-1;

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top()>ar[i])

        {

            output[i]=st.top();

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top()<=ar[i])

        {

            while(!st.empty() && st.top()<=ar[i])

            {

                st.pop();

            }

            if(st.empty())

            {

                output[i]=-1;

            }

            else

            {

                output[i]=st.top();

            }

            st.push(ar[i]);

        }

    }

    for ( i = 0; i < n; i++)

    {

        cout<<output[i]<<'\t';

    }

    return 0;

}

**NEXT SMALLER RIGHT**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st;

    int n,i;

    cout<<"enter no. of elements"<<endl;

    cin>>n;

    int ar[n],out[n];

    memset(out,n,0);

    cout<<"enter the values"<<endl;

    for(i=0;i<n;i++)

    {

        cin>>ar[i];

    }

    for ( i = n-1; i >=0; i--)

    {

        if(st.empty())

        {

            st.push(ar[i]);

            out[i]=-1;

        }

        else if(!st.empty() && st.top()<ar[i])

        {

            out[i]=st.top();

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top()>=ar[i])

        {

            while(!st.empty() && st.top()>=ar[i])

            {

                st.pop();

            }

            if(st.empty())

            {

                out[i]=-1;

            }

            else

            {

                out[i]=st.top();

            }

            st.push(ar[i]);

        }

    }

    for ( i = 0; i < n; i++)

    {

        cout<<out[i]<<'\t';

    }

    return 0;

}

**NEXT GREATER LEFT**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st;

    int n,i;

    cout<<"enter total values"<<endl;

    cin>>n;

    int ar[n],out[n];

    for(i=0;i<n;i++)

    {

        cin>>ar[i];

    }

    memset(out,n,0);

    for(i=0;i<n;i++)

    {

        if(st.empty())

        {

            out[i]=-1;

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top()>ar[i])

        {

            out[i]=st.top();

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top()<=ar[i])

        {

            while(!st.empty() && st.top()<=ar[i])

            {

                st.pop();

            }

            if(st.empty())

            {

                out[i]=-1;

            }

            else

            {

                out[i]=st.top();

            }

            st.push(ar[i]);

        }

    }

    for ( i = 0; i < n; i++)

    {

        cout<<out[i]<<'\t';

    }

    return 0;

}

**NEXT SMALLER LEFT**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st;

    int n,i;

    cout<<"enter total values"<<endl;

    cin>>n;

    int out[n],ar[n];

    memset(out,n,0);

    cout<<"enter the values"<<endl;

    for ( i = 0; i < n; i++)

    {

        cin>>ar[i];

    }

    for ( i = 0; i < n; i++)

    {

        if(st.empty())

        {

            out[i]=-1;

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top() < ar[i])

        {

            out[i]=st.top();

            st.push(ar[i]);

        }

        else if(!st.empty() && st.top() >= ar[i])

        {

            while(!st.empty() && st.top() >= ar[i])

            {

                st.pop();

            }

            if(st.empty())

            {

                out[i]=-1;

            }

            else

            {

                out[i]=st.top();

            }

            st.push(ar[i]);

        }

    }

    for ( i = 0; i < n; i++)

    {

        cout<<out[i]<<'\t';

    }

    return 0;

}

**STOCK SPAN**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st,index;

    int n,i;

    cout<<"enter no. of inputs"<<endl;

    cin>>n;

    int out[n],ar[n];

*//memset(out,n,0);*

    for (i = 0; i < n; i++)

    {

        cin>>ar[i];

    }

    for(i=0;i<n;i++)

    {

        if(st.empty())

        {

            st.push(ar[i]);

            out[i]=1;

            index.push(i);

        }

        else if(!st.empty() && st.top()>ar[i])

        {

            out[i]=i-index.top();

            st.push(ar[i]);

            index.push(i);

        }

        else if(!st.empty() && st.top()<=ar[i])

        {

            while(!st.empty() && st.top()<=ar[i])

            {

                st.pop();

                index.pop();

            }

            if(st.empty())

            {

                out[i]=1;

            }

            else

            {

                out[i]=i-index.top();

            }

            st.push(ar[i]);

            index.push(i);

        }

*//cout<<"pos is "<<pos<<endl;*

    }

    for ( i = 0; i < n; i++)

    {

        cout<<out[i]<<'\t';

    }

    return 0;

}

**CHECK BALANCED PARANTHESIS**

#include <bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st;

    int i,n,check=1;

    string str;

    cin>>str;

    n=str.length();

    for(i=0;i<n;i++)

    {

        if(str[i]=='{' || str[i]=='(' || str[i]=='[' )

        {

            st.push(str[i]);

        }

        else if(str[i]=='}')

        {

            if(st.top()=='{')

            {

                st.pop();

            }

            else

            {

                check =0;

            }

        }

        else if(str[i]==')')

        {

            if(st.top()=='(')

            {

                st.pop();

            }

            else

            {

                check =0;

            }

        }

        else if(str[i]==']')

        {

            if(st.top()=='[')

            {

                st.pop();

            }

            else

            {

                check= 0;

            }

        }

        else if(st.empty())

        {

            check= 0;

        }

    }

    if(check==1)

    {

        cout<<"balanced";

    }

    else

    {

        cout<<"not balanced";

    }

    return 0;

}

**LARGEST AREA RECTANGLE IN HISTOGRAM**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    stack<int> st1,st2,index\_left,index\_right;

    int n,i;

    cout<<"enter no. of inputs"<<endl;

    cin>>n;

    int ar[n],out\_left[n],out\_right[n];

    for(i=0;i<n;i++)

    {

        cin>>ar[i];

    }

    for(i=0;i<n;i++)

    {

        if(st1.empty())

        {

            st1.push(ar[i]);

            index\_left.push(i);

            out\_left[i]=index\_left.top();

        }

        else if(!st1.empty() && st1.top()<ar[i])

        {

            st1.push(ar[i]);

            out\_left[i]=index\_left.top();

            index\_left.push(i);

        }

        else if(!st1.empty() && st1.top()>=ar[i])

        {

            while(!st1.empty() && st1.top()>=ar[i])

                {

                    st1.pop();

                    index\_left.pop();

                }

            if(st1.empty())

            {

                index\_left.push(i);

                out\_left[i]=index\_left.top();

            }

            else

            {

                out\_left[i]=index\_left.top();

                index\_left.push(i);

            }

        }

    }

    for ( i = 0; i < n; i++)

    {

        cout<<out\_left[i]<<'\t';

    }

    return 0;

}

**RAIN WATER TAPPING**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    int n,val,sum=0;

    cout<<"enter no. of inputs"<<endl;

    cin>>n;

    int ar[n],i,left[n],right[n];

    cout<<"enter the values"<<endl;

    for(i=0;i<n;i++)

    {

        cin>>ar[i];

        if(i==0)

        {

            left[i]=ar[i];

        }

        else

        {

            if(left[i-1]<ar[i])

            {

                left[i]=ar[i];

            }

            else

            {

                left[i]=left[i-1];

            }

        }

    }

    for(i=n-1;i>=0;i--)

    {

        if(i==n-1)

        {

            right[i]=ar[i];

        }

        else

        {

            if(right[i+1]<ar[i])

            {

                right[i]=ar[i];

            }

            else

            {

                right[i]=right[i+1];

            }

        }

    }

    for(i=0;i<n;i++)

    {

        val=(left[i]<=right[i]?left[i]:right[i]);

        if(val>ar[i])

        {

            sum=sum+(val-ar[i]);

        }

    }

    cout<<"the value is "<<sum<<endl;

    return 0;

}