

BST

```
#include <bits/stdc++.h>

using namespace std;

class Node
{
public:
    int val;
    Node *left;
    Node *right;
    Node(int val)
    {
        this->val = val;
        this->left = NULL;
        this->right = NULL;
    }
};

void insert_in_bst(Node *&root, int val)
{
    Node *newnode = new Node(val);
    if (root == NULL)
    {
        root = newnode;
        return;
    }

    if (root->val > val)
    {
        insert_in_bst(root->left, val);
    }
}
```

```
    else
    {
        insert_in_bst(root->right, val);
    }
}
```

```
void inorder(Node *root)
{
    if (root == NULL)
    {
        return;
    }
    inorder(root->left);
    cout << root->val << endl;
    inorder(root->right);
}
```

```
bool search_value(Node *root, int val)
{
    if (root == NULL)
    {
        return false;
    }
```

```
    if (root->val == val)
        return true;
```

```
    if (root->val > val)
        return search_value(root->left, val);
    else
```

```
        return search_value(root->right, val);  
    }  
}
```

```
int main()
```

```
{
```

```
    Node *root = NULL;
```

```
    int val;
```

```
    while (true)
```

```
    {
```

```
        cin >> val;
```

```
        if (val == -1)
```

```
        {
```

```
            break;
```

```
        }
```

```
        insert_in_bst(root, val);
```

```
    }
```

```
    inorder(root);
```

```
    if (search_value(root, 20))
```

```
        cout << "Yes" << endl;
```

```
    else
```

```
        cout << "NO" << endl;
```

```
    return 0;
```

```

-1  

2      break;  

4  

5    }  

14  

20    insert_in_bst(root, val);  

21  

26  }  

Yes  
  

Process returned 0 (0x0)   execution time : 186.896 s  

Press any key to continue.  

        if (search_value(root, 20))  

}

```

Singly Linked list

```
#include <bits/stdc++.h>

using namespace std;

class Node
{
public:
    int val;
    Node *next;
    Node(int val)
    {
        this->val = val;
        this->next = NULL;
    }
};

void insert_tail(Node *&head, Node *&tail, int val)
{
    Node *newNode = new Node(val);
    if (head == NULL)
    {
```

```

        head = newNode;
        tail = newNode;
        return;
    }
    tail->next = newNode;
    tail = newNode;
}

void print_linked_list(Node *head)
{
    Node *tmp = head;
    cout << "Your created link list" << endl;
    while (tmp != NULL)
    {
        cout << tmp->val << " ";
        tmp = tmp->next;
    }
    cout << endl;
}

int main()
{
    Node *head = NULL;
    Node *tail = NULL;
    int val;
    cout << "Input your singly link list" << endl;
    cout << "For terminated input enter -1" << endl;
    while (true)
    {
        cin >> val;
        if (val == -1)
            break;

```

```

        insert_tail(head, tail, val);
    }
    print_linked_list(head);
    return 0;

```

```

Input your singly link list
For terminated input enter -1
20 1 0 5 36
-1
Your created link list
20 1 0 5 36
main()
Process returned 0 (0x0)   execution time : 8.138 s
Press any key to continue.
Node *head = NULL;
Node *tail = NULL;
int val;
}

```

Stack

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
class Node
```

```
{
```

```
public:
```

```
    int val;
```

```
    Node *prev;
```

```
    Node *next;
```

```
    Node(int val)
```

```
{
```

```
    this->val = val;
```

```
    this->next = NULL;
```

```
    this->prev = NULL;
```

```
}
```

```
};
```

```
class Mystack
```

```
{
```

public:

Node *head = NULL;

Node *tail = NULL;

int sz = 0;

void push(int val)

{

sz++;

Node *newnode = new Node(val);

if (head == NULL)

{

head = newnode;

tail = newnode;

return;

}

tail->next = newnode;

newnode->prev = tail;

tail = newnode;

}

void pop()

{

if (tail == NULL)

{

cout << "Stack is empty" << endl;

return;

}

sz--;

Node *deletenode = tail;

tail = tail->prev;

```
if (tail == NULL)
{
    head = NULL;
}
else
{
    tail->next = NULL;
}
```

```
delete deletenode;
}
```

```
int top()
{
    if (tail != NULL)
    {
        return tail->val;
    }

    else
    {
        cout << "Stack is empty" << endl;
        return -1;
    }
}
```

```
int size()
{
    return sz;
}
```



```

    }

    bool empty()
    {
        if (sz == 0)
            return true;
        else
            return false;
    }
};

int main()
{
    Mystack st;

    int n;
    cout << "Enter your stack size" << endl;
    cin >> n;
    cout << "Given your number:" << endl;

    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        st.push(x);
    }
    cout << "Your stack" << endl;
    while (!st.empty())
    {
        cout << st.top() << " ";
    }
}

```

```

        st.pop();
    }
    cout << endl;
    return 0;
}

```

```

Enter your stack size
5
Your number:" << endl;
Given your number:
12 13 14 15 16
Your stack
16 15 14 13 12

Process returned 0 (0x0)   execution time : 30.535 s
Press any key to continue.
stack" << endl;
pty())

```

Bubble Sort

```

#include <bits/stdc++.h>

using namespace std;

int main()
{

    int n;

    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)

```

```

    {

        if (arr[j] > arr[j + 1])
        {
            swap(arr[j], arr[j + 1]);
        }
    }
}

for (int i = 0; i < n; i++)
{
    cout << arr[i] << " ";
}

return 0;
}

```

```

5
12
20
33
25
64
12 20 25 33 64
Process returned 0 (0x0)   execution time : 7.522 s
Press any key to continue.
swap(arr[j], arr[j + 1]);

```

Insertion Sort

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n;

```

```

cin >> n;

int arr[n];

for (int i = 0; i < n; i++)
{
    cin >> arr[i];
}

int j;

for (int i = 0; i < n; i++)
{
    j=i;
    while (j>0 && arr[j-1]>arr[j])
    {
        swap(arr[j],arr[j-1]);
        j=j-1;
    }
}

for (int i = 0; i < n; i++)
{
    cout<< arr[i]<<" ";
}

return 0;
}

```

```

5
10 2 1 4 5
1 2 4 5 10
Process returned 0 (0x0)   execution time : 6.994 s
Press any key to continue.

```

Selection Sort:

```

#include <bits/stdc++.h>

using namespace std;

```

```
int main()
{
    int n;
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    for (int i = 0; i < n - 1; i++)
    {
        int min_index = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[min_index])
            {
                min_index = j;
            }
            if (min_index != i)
            {
                swap(arr[i], arr[min_index]);
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    return 0;
}
```

```
5
0 14 3 9 4
0 3 4 9 14
Process returned 0 (0x0)   execution time : 9.928 s
Press any key to continue.
#include <bits/stdc++.h>
using namespace std;
int main()
```