

NAME : MASOOD SAID

ID : 13723

Subject : data structure and algorithms

Major Assignment lab work

Instructor : Muhammad Adil

Bubblesort

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  void swap(int *xp, int *yp)
5  {
6      int temp = *xp;
7      *xp = *yp;
8      *yp = temp;
9  }
10
11 void bubbleSort(int arr[], int n)
12 {
13     int i, j;
14     for (i = 0; i < n-1; i++)
15
16         for (j = 0; j < n-i-1; j++)
17             if (arr[j] > arr[j+1])
18                 swap(&arr[j], &arr[j+1]);
19 }
20
21
22 void printArray(int arr[], int size)
23 {
24     int i;
25     for (i = 0; i < size; i++)
26         cout << arr[i] << " ";
27     cout << endl;
28 }
29
30
31 int main()
32 {
33     int arr[] = {64, 34, 25, 12, 22, 11, 90};
34     int n = sizeof(arr)/sizeof(arr[0]);
35     bubbleSort(arr, n);
36     cout<<"Sorted array: \n";
37     printArray(arr, n);
38     return 0;
39 }
40
```

OUTPUT

```
Sorted array:  
11 12 22 25 34 64 90  
  
-----  
Process exited after 0.01313 seconds with return value 0  
Press any key to continue . . .
```

InsertionSort

```
2  #include <bits/stdc++.h>
3  using namespace std;
4
5
6  void insertionSort(int arr[], int n)
7  {
8      int i, key, j;
9      for (i = 1; i < n; i++)
10     {
11         key = arr[i];
12         j = i - 1;
13
14
15         while (j >= 0 && arr[j] > key)
16         {
17             arr[j + 1] = arr[j];
18             j = j - 1;
19         }
20         arr[j + 1] = key;
21     }
22 }
23
24
25 void printArray(int arr[], int n)
26 {
27     int i;
28     for (i = 0; i < n; i++)
29         cout << arr[i] << " ";
30     cout << endl;
31 }
32
33
34 int main()
35 {
36     int arr[] = { 12, 11, 13, 5, 6 };
37     int n = sizeof(arr) / sizeof(arr[0]);
38
39     insertionSort(arr, n);
40     printArray(arr, n);
41
42     return 0;
43 }
44
```

Output

```
5 6 11 12 13
```

```
-----  
Process exited after 0.01286 seconds with return value 0  
Press any key to continue . . .
```

SelectionSort

```
3
4 void swap(int *xp, int *yp)
5 {
6     int temp = *xp;
7     *xp = *yp;
8     *yp = temp;
9 }
10
11 void selectionSort(int arr[], int n)
12 {
13     int i, j, min_idx;
14
15     for (i = 0; i < n-1; i++)
16     {
17         min_idx = i;
18         for (j = i+1; j < n; j++)
19             if (arr[j] < arr[min_idx])
20                 min_idx = j;
21
22         swap(&arr[min_idx], &arr[i]);
23     }
24 }
25
26 void printArray(int arr[], int size)
27 {
28     int i;
29     for (i=0; i < size; i++)
30         cout << arr[i] << " ";
31     cout << endl;
32 }
33
34 int main()
35 {
36     int arr[] = {64, 25, 12, 22, 11};
37     int n = sizeof(arr)/sizeof(arr[0]);
38     selectionSort(arr, n);
39     cout << "Sorted array: \n";
40     printArray(arr, n);
41     return 0;
42 }
```

Output

```
Sorted array:  
11 12 22 25 64  
  
-----  
Process exited after 0.01106 seconds with return value 0  
Press any key to continue . . .
```

Binary search tree

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4
5 int binarySearch(int arr[], int l, int r, int x)
6 {
7     if (r >= l) {
8         int mid = l + (r - l) / 2;
9
10        if (arr[mid] == x)
11            return mid;
12
13        if (arr[mid] > x)
14            return binarySearch(arr, l, mid - 1, x);
15
16        return binarySearch(arr, mid + 1, r, x);
17    }
18
19    // We reach here when element is not
20    // present in array
21    return -1;
22 }
23
24
25
26 int main(void)
27 {
28     int arr[] = { 2, 3, 4, 10, 40 };
29     int x = 10;
30     int n = sizeof(arr) / sizeof(arr[0]);
31     int result = binarySearch(arr, 0, n - 1, x);
32     (result == -1) ? cout << "Element is not present in array"
33                   : cout << "Element is present at index " << result;
34     return 0;
35 }
36
```


Output

```
Element is present at index 3  
-----  
Process exited after 0.01898 seconds with return value 0  
Press any key to continue . . .
```