

**OPERATING SYSTEMS FILE**

**Name: Aneesh Kundu**

**Roll Number: 2K17/CO/054**

**Batch: COE/A1**

**Group 3**

**Submitted To: Daya Gupta Mam**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Aim of Program | Pages | Date | Sign |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

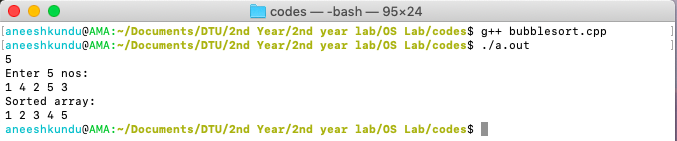
**AIM: Program for Bubble Sort on Linux Operating System**

**Theory:** In this program we aim to sort an array using Bubble Sort Algorithm. Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

**Code:**

1. #include <iostream>
2. using namespace std;
3. void bubblesort(int\* a,int n){
4. for(int i = 0; i < n - 1; i++){
5. for(int j = 0; j < n - i - 1; j++){
6. if(a[j] > a[j + 1])
7. swap(a[j + 1],a[j]);
8. }
9. }
10. }
11. int main(){
12. int n,a[20];
13. cin>>n;
14. cout<<"Enter "<<n<<" nos:\n";
15. for(int i = 0; i < n ;i++)
16. cin>>a[i];
17. bubblesort(a,n);
18. cout<<"Sorted array:\n";
19. for(int i = 0; i < n ;i++)
20. cout<<a[i]<<" ";
21. cout<<endl;
22. return 0;
23. }

**Output:**



**Discussion:**

The Time Complexity of the above program is:

**Worst and Average Case Time Complexity:**O(n\*n). Worst case occurs when array is reverse sorted.

**Best Case Time Complexity:** O(n). Best case occurs when array is already sorted.

The Space Complexity of the above program is:

Auxiliary Space: O(1)