AOA PRACTICAL LAB 1

Name: Brendan Lucas, Roll No:8953, Div: SE Comp B

Source Code:

```
#include <iostream>
#include <cstdlib>
using namespace std;

/int*/void bubblesort(int *arr,int n)
{
	// Declaring variables
	int j,i,swap,temp;

	//First for loop
	for(i=0;i<n-1;i++)
	{
		//setting swap to 0
		swap=0;

		//Second for loop
		for(j=0;j<n-i-1;j++)
		{
			//If current element is greater than the next element
			if(arr[j]>arr[j+1])
			{
				temp=arr[j];
				arr[j]=arr[j+1];
				arr[j+1]=temp;
				swap++;
			}
		}

		//If no swapping is done i.e. array is sorted, then break
		if(swap==0)
		{
			break;
		}
	}
//	return arr;
}
int main(void)

{

	//Declering variables
	int n,*arr,i;

	//Asking for No. of elements in array
	cout<<"Enter the no of elements in array:-\n";
	cin>>n;

	//Creating array of given size using malloc
	arr=(int*)malloc(sizeof(int)*n);

	//Taking input of elements of array
	cout<<"Enter the array\n";
	for(i=0;i<n;i++)
	{
		cin>>arr[i];
	}

	//Sorting the array
	bubblesort(arr, n);
	//selectionsort(arr, n);
	//quicksort(arr,0,n-1);

	//Displaying the final result
	cout<<"\nThe Sorted array is:- \n";
	for(i=0;i<n;i++)
	{
		cout<<arr[i]<<" ";
	}
	return 0;
}
```

Page No.
Date

Modified Bubble Sort :-
code :-

```
for (i=0; i<n; i++)
{
    swap = 0
    for (j=0; j<n-i-1; j++)
    {
        if (arr[j] > arr[j+1])
        {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
            swap++;
        }
    }
    if (swap == 0)
    {
        break;
    }
}
```

Time analysis :-
Worst case :-
when ~~Values~~ ~~arr~~ are in reverse order :- $O(n^2)$
(same as normal bubble sort)

Best case :-
when values are in ascending order :- $O(n)$
(As the values are in ascending order i.e
in sorted order, There will be no swapping.
Hence the array will be traversed ~~Once~~ Once and then Stop)