

Title: Write a program to implement Parallel Bubble Sort using OpenMP

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
using namespace std;

void bubble(int *, int);
void swap(int &, int &);

void bubble(int *a, int n)
{
    for( int i = 0; i < n; i++ )
    {
        int first = i % 2;

        #pragma omp parallel for shared(a,first)
        for( int j = first; j < n-1; j += 2 )
        {
            if( a[ j ] > a[ j+1 ] )
            {
                swap( a[ j ], a[ j+1 ] );
            }
        }
    }
}

void swap(int &a, int &b)
{
    int test;
    test=a;
    a=b;
    b=test;
}

int main()
{
```

```

int *a,n;
cout<<"\n enter total no of elements=>";
cin>>n;
a=new int[n];
cout<<"\n enter elements=>";
for(int i=0;i<n;i++)
{
    cin>>a[i];
}

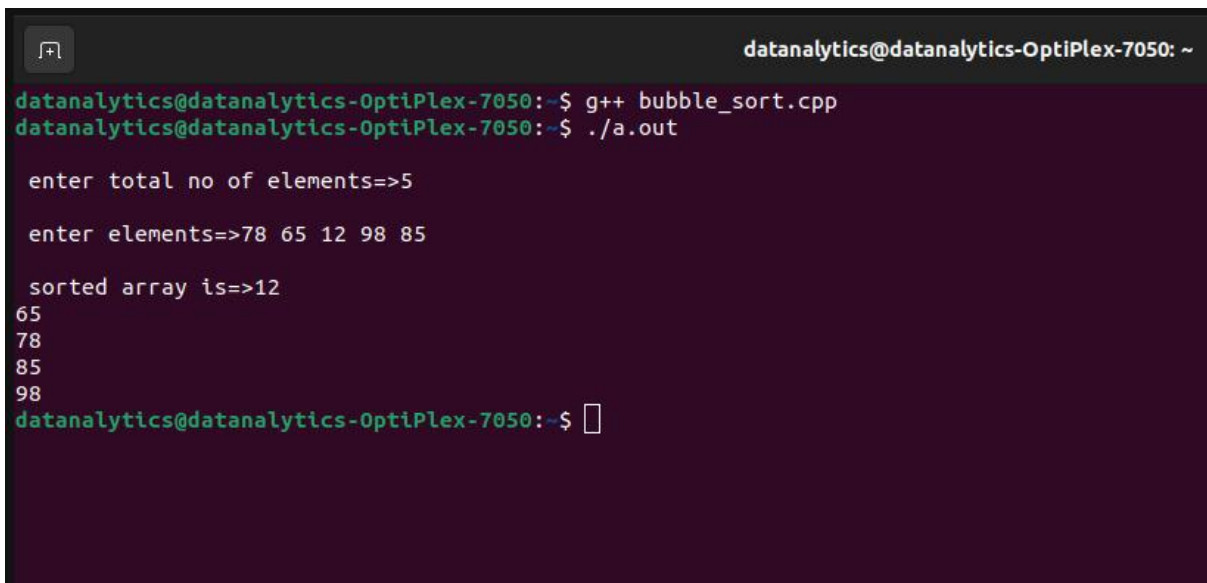
bubble(a,n);

cout<<"\n sorted array is=>";
for(int i=0;i<n;i++)
{
    cout<<a[i]<<endl;
}

return 0;
}

```

Output:



```

datanalytics@datanalytics-OptiPlex-7050: ~
datanalytics@datanalytics-OptiPlex-7050:~$ g++ bubble_sort.cpp
datanalytics@datanalytics-OptiPlex-7050:~$ ./a.out

enter total no of elements=>5

enter elements=>78 65 12 98 85

sorted array is=>12
65
78
85
98
datanalytics@datanalytics-OptiPlex-7050:~$ 

```