Assignment 2(b)

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;
}
```

How to Run code in Ubuntu:

1. **Open a terminal window.**

Compile the code using the following command:

```
g++ -fopenmp filename.cpp -o executable
```

Run the program by executing the following command:

```bash
./executable
```

Output

```c
Sorted array: 1 2 3 4 5 6 7 8 9
```

This is because the input array `{5, 3, 1, 9, 8, 2, 4, 7, 6}` is sorted in ascending order using the parallel bubble sort algorithm implemented in the `parallel_bubble_sort()` function. The sorted array is then printed to the console in the `main()` function using a for loop.