

Name: Manasi. B.Kshirsagar

PRN: ST17111009

Roll No.: 06

Class: BE Comp SS

Code (Bubble Sort):

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
#include <cstdlib> // For srand() and rand()
#include <ctime>   // For time()
#include<chrono>
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()
{
    chrono::time_point<chrono::system_clock> start, end;
    start = chrono::system_clock::now();
    int *a,n;

    srand(time(0)); // Initialize random number generator.

    cout<<"\n Enter total no of elements=>";
}

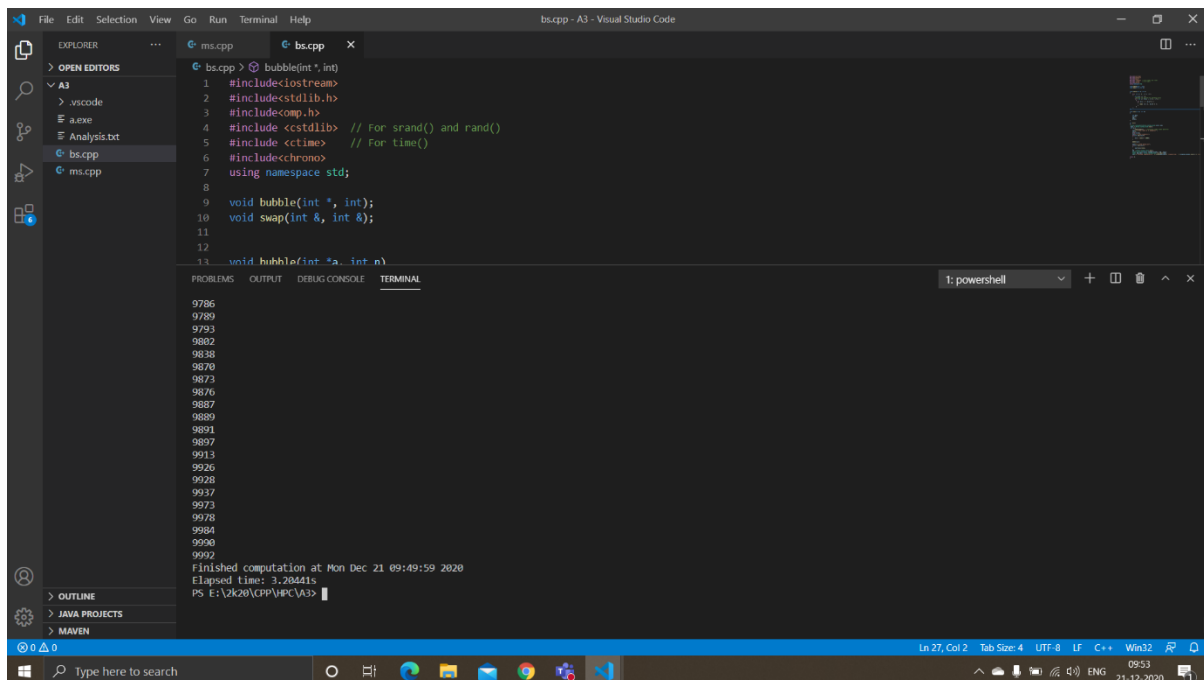
```

```

cin>>n;
a=new int[n];
cout<<"\n enter elements=>";
for(int i=0;i<n;i++)
{
    a[i] = (rand() % 10000);
}
bubble(a,n);
cout<<"\n Sorted array is=>";
for(int i=0;i<n;i++)
{
    cout<<a[i]<<endl;
}
end = chrono::system_clock::now();
chrono::duration<double> elapsed_seconds = end - start;
time_t end_time = chrono::system_clock::to_time_t(end);
cout << "Finished computation at " << ctime(&end_time)<< "Elapsed
time: " << elapsed_seconds.count() << "s\n";
return 0;
}

```

Output:



The screenshot shows the Visual Studio Code interface with a C++ file named `bs.cpp` open. The code defines a `bubble` function and a `swap` function, then calls `void bubble(int *a, int n)`. The terminal window shows the output of the program, which is a list of numbers: 9786, 9780, 9793, 9802, 9838, 9870, 9873, 9876, 9887, 9889, 9891, 9897, 9913, 9926, 9928, 9937, 9973, 9978, 9984, 9990, 9992. The terminal also shows the completion time: "Finished computation at Mon Dec 21 09:49:59 2020" and "Elapsed time: 3.28441s".

```
bs.cpp > bubble(int *, int)
1  #include<iostream>
2  #include<stdlib.h>
3  #include<omp.h>
4  #include <cstdlib> // For srand() and rand()
5  #include <ctime>   // For time()
6  #include<chrono>
7  using namespace std;
8
9  void bubble(int *, int);
10 void swap(int &, int &);
11
12
13 void bubble(int *a, int n)
```

9786  
9780  
9793  
9802  
9838  
9870  
9873  
9876  
9887  
9889  
9891  
9897  
9913  
9926  
9928  
9937  
9973  
9978  
9984  
9990  
9992

Finished computation at Mon Dec 21 09:49:59 2020  
Elapsed time: 3.28441s  
PS E:\2k20\CPP\VC\A3>

Code (Merge Sort):

```
#include <iostream>
```

```
#include <stdlib.h>
```

```
#include <omp.h>
```

```
#include <cstdlib> // For srand() and rand()
```

```
#include <ctime>   // For time()
```

```
#include <chrono>
```

```
using namespace std;
```

```
void mergesort(int a[],int i,int j,int n);
```

```
void merge(int a[],int i1,int j1,int i2,int j2,int n);
```

```
void mergesort(int a[],int i,int j,int n)
```

```

{
    int mid;
    if(i<j)
    {
        mid=(i+j)/2;
        //#pragma omp parallel sections
        //{
        // #pragma omp section
        {
            mergesort(a,i,mid,n);
        }
        // #pragma omp section
        {
            mergesort(a,mid+1,j,n);
        }
        //}
        merge(a,i,mid,mid+1,j,n);
    }
}

```

```

void merge(int a[],int i1,int j1,int i2,int j2,int n)
{
    int temp[n];
    int i,j,k;
    i=i1;
    j=i2;

```

```
k=0;
```

```
while(i<=j1 && j<=j2)
```

```
{
```

```
    if(a[i]<a[j])
```

```
    {
```

```
        temp[k++]=a[i++];
```

```
    }
```

```
    else
```

```
    {
```

```
        temp[k++]=a[j++];
```

```
    }
```

```
}
```

```
while(i<=j1)
```

```
{
```

```
    temp[k++]=a[i++];
```

```
}
```

```
while(j<=j2)
```

```
{
```

```
    temp[k++]=a[j++];
```

```
}
```

```
for(i=j1,j=0;i<=j2;i++,j++)
```

```
{
```

```
    a[i]=temp[j];
```

```
}  
}
```

```
int main()
```

```
{
```

```
    chrono::time_point<chrono::system_clock> start, end;
```

```
    start = chrono::system_clock::now();
```

```
    int *a,n,i;
```

```
    cout<<"\n Enter total No of Elements: => ";
```

```
    cin>>n;
```

```
    a= new int[n];
```

```
    cout<<"\n Entering Random Elements in array =>";
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        a[i] = (rand() % 10000);
```

```
    }
```

```
    mergesort(a, 0, n-1,n);
```

```
    cout<<"\n Sorted Array is=>";
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        cout<<a[i]<<endl;
```

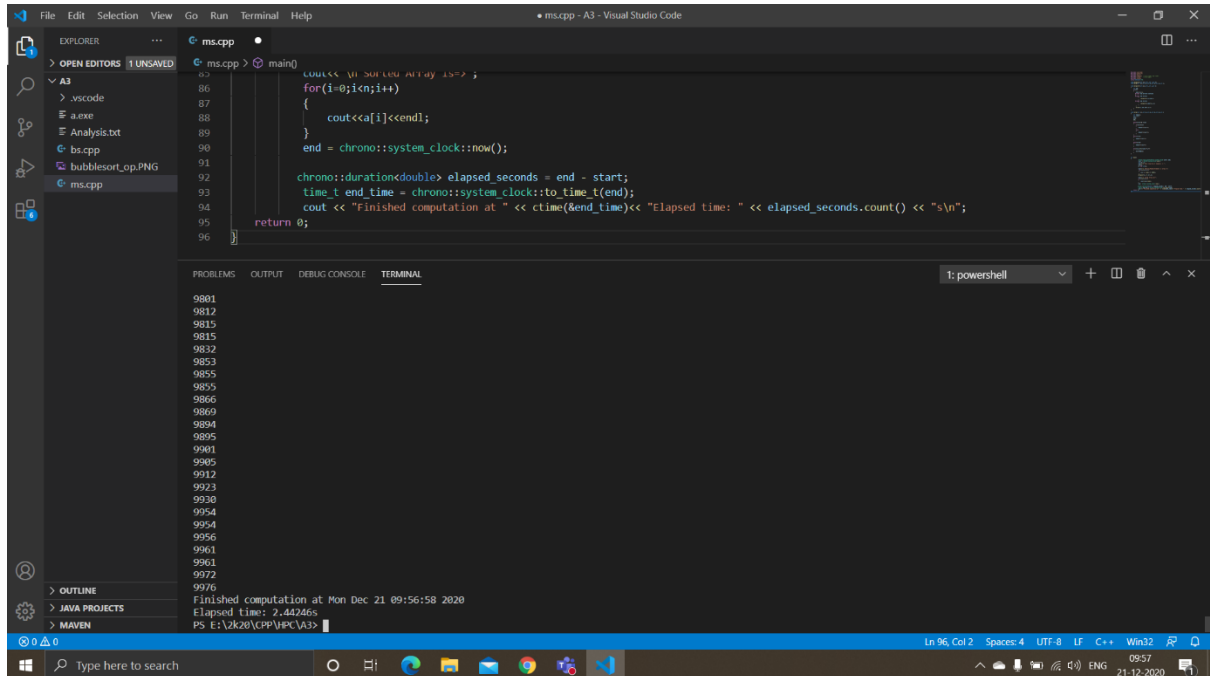
```
    }
```

```
    end = chrono::system_clock::now();
```

```
    chrono::duration<double> elapsed_seconds = end - start;
    time_t end_time = chrono::system_clock::to_time_t(end);
    cout << "Finished computation at " << ctime(&end_time)<<
    "Elapsed time: " << elapsed_seconds.count() << "s\n";
    return 0;
}
```



Output:



The screenshot shows the Visual Studio Code interface with a C++ file named `ms.cpp` open. The code in the editor is as follows:

```
1 // Sort the array in-2 ;
2
3 #include <iostream>
4 #include <chrono>
5
6 using namespace std;
7
8 int main()
9 {
10     cout<<"The Sorted Array is:-";
11     for(i=0;i<n;i++)
12     {
13         cout<<a[i]<<endl;
14     }
15     end = chrono::system_clock::now();
16
17     chrono::duration<double> elapsed_seconds = end - start;
18     time_t end_time = chrono::system_clock::to_time_t(end);
19     cout << "Finished computation at " << ctime(&end_time)<< "Elapsed time: " << elapsed_seconds.count() << "s\n";
20
21     return 0;
22 }
```

The terminal output shows the execution of the program, displaying the sorted array and the elapsed time:

```
9801
9812
9815
9815
9815
9832
9853
9855
9855
9866
9869
9894
9895
9901
9905
9912
9923
9930
9954
9954
9956
9961
9961
9972
9976
Finished computation at Mon Dec 21 09:56:58 2020
Elapsed time: 2.44266s
Ps E:\2k20\CPP\HPC\A3>
```

The status bar at the bottom indicates the current line and column (Ln 96, Col 2), the encoding (UTF-8), the language (C++), and the platform (Win32). The system clock shows 09:57 on 21-12-2020.