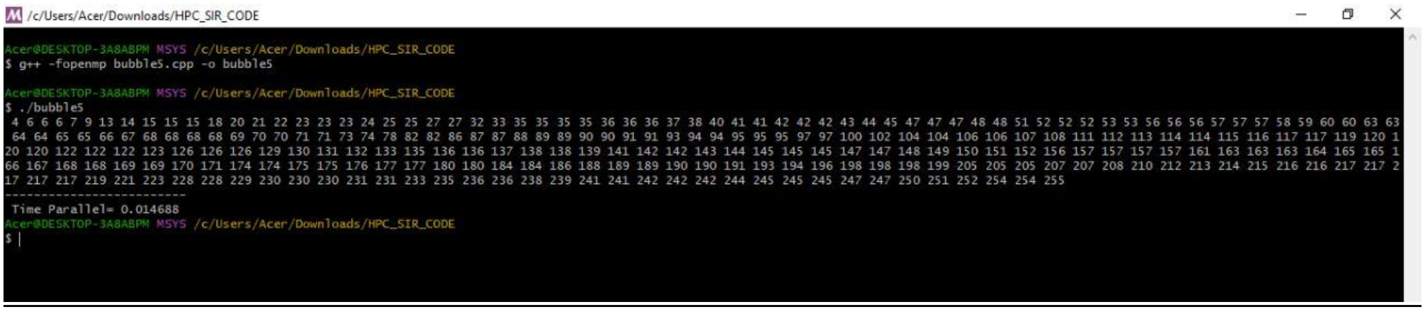
A - Merge sort

CODE:

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
using namespace std;
void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);
void mergesort(int a[],int i,int j)
{
  int mid;
  if(i<j)
  {
    mid=(i+j)/2;
    #pragma omp parallel sections
    {
      #pragma omp section
     {
        mergesort(a,i,mid);
      }
      #pragma omp section
      {
        mergesort(a,mid+1,j);
      }
    }
    merge(a,i,mid,mid+1,j);
  }
}
void merge(int a[],int i1,int j1,int i2,int j2)
{
  int temp[1000];
  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=i1,j=0;i<=j2;i++,j++)
  {
    a[i]=temp[j];
  }
}
int main()
{
  int *a,n,i;
  cout<<"\n enter total no of elements=>";
  cin>>n;
  a= new int[n];
  cout<<"\n enter elements=>\n";
  for(i=0;i<n;i++)
  {
    cin>>a[i];
  }
  mergesort(a, 0, n-1);
  cout<<"\n sorted array is=>";
  for(i=0;i<n;i++)
  {
    cout<<"\n"<<a[i];
  }
  return 0;
}
```
OUTPUT :



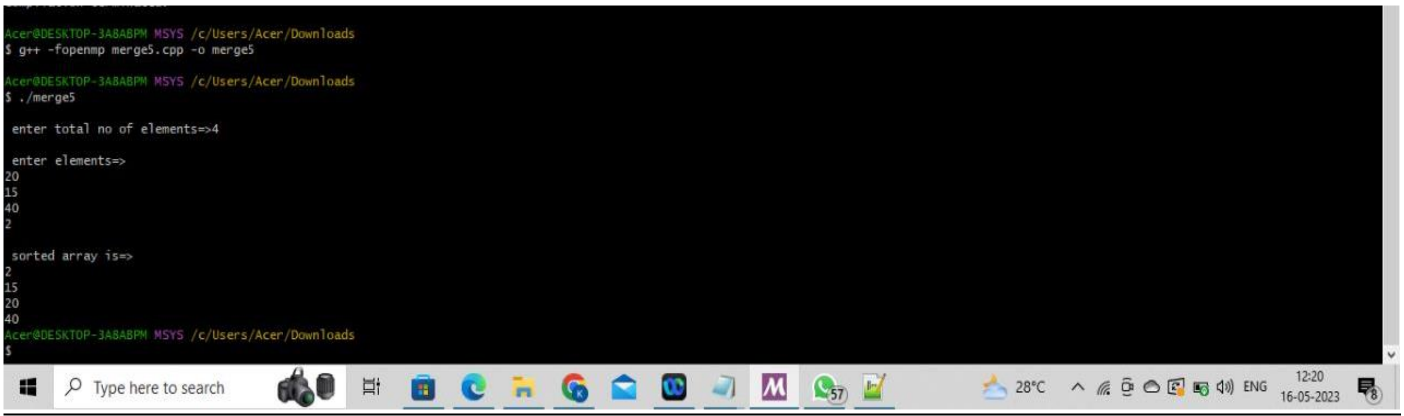
```
/c/Users/Acer/Downloads/HPC_SIR_CODE

Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads/HPC_SIR_CODE
$ g++ -fopenmp bubble5.cpp -o bubble5

Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads/HPC_SIR_CODE
$ ./bubble5
 4 6 6 6 7 9 13 14 15 15 15 18 20 21 22 23 23 23 24 25 25 27 27 32 33 35 35 35 35 36 36 36 37 38 40 41 41 42 42 42 43 44 45 47 47 47 48 48 51 52 52 52 53 53 56 56 56 57 57 57 58 59 60 60 63 63
 64 64 65 65 66 67 68 68 68 68 69 70 70 71 71 73 74 78 82 82 86 87 87 88 89 89 90 90 91 91 93 94 94 95 95 95 97 97 100 102 104 104 106 106 107 108 111 112 113 114 114 115 116 117 117 119 120 1
20 120 122 122 122 123 126 126 126 129 130 131 132 133 135 136 136 137 138 138 139 141 142 142 143 144 145 145 145 147 147 148 149 150 151 152 156 157 157 157 157 161 163 163 163 164 165 165 1
66 167 168 168 169 169 170 171 174 174 175 175 176 177 177 180 180 184 184 186 188 189 189 190 190 191 193 194 196 198 198 198 199 205 205 205 207 207 208 210 212 213 214 215 216 216 217 217 2
17 217 217 219 221 223 228 228 229 230 230 230 231 231 233 235 236 236 238 239 241 241 242 242 242 244 245 245 245 247 247 250 251 252 254 254 255
-------------------------
 Time Parallel= 0.014688
Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads/HPC_SIR_CODE
$ |
```

<u>B – Bubble Sort</u>

<u>CODE:</u>

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

void swap(int *num1, int *num2);

int main (int argc, char *argv[]) {
        int SIZE =1<<8;
        int A[SIZE];
        for(int i=0;i<SIZE;i++)
        {
          A[i]=rand()%SIZE;
        }
        //int A[5] = {6,9,1,3,7};
        int N = SIZE;
        int i=0, j=0;
        int first;
        double start,end;
        start=omp_get_wtime();
        for( i = 0; i < N-1; i++ )
        {
                first = i % 2;
                #pragma omp parallel for default(none),shared(A,first,N)
                for( j = first; j < N-1; j += 1 )
                {
                        if( A[ j ] > A[ j+1 ] )
                        {
                                swap( &A[ j ], &A[ j+1 ] );
                        }
                }
        }
end=omp_get_wtime();
        for(i=0;i<N;i++)
        {
                printf(" %d",A[i]);
        }

printf("\n-------------------------\n Time Parallel= %f",(end-start));
}

void swap(int *num1, int *num2)
{

        int temp = *num1;
        *num1 =  *num2;
        *num2 = temp;
}
```

Output :



```
Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads
$ g++ -fopenmp merge5.cpp -o merge5

Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads
$ ./merge5

 enter total no of elements=>4

 enter elements=>
20
15
40
2

 sorted array is=>
2
15
20
40
Acer@DESKTOP-3A8ABPM MSYS /c/Users/Acer/Downloads
$
```