HOMEWORK 9

SORTING

1. (20 points) Create a program to compute the median of an array consisting of N integers!

Source Code

```
#include <iostream>

using namespace std;

void insertionSort(int data[], int dataSize){
   int i, temp;

   for(int j = 1; j<dataSize; j++) {
      i = j-1;
      temp=data[j];
      while(data[i]>temp && i>=0){
         data[i+1] = data[i];
         i--;
      }
      data[i+1] = temp;
   }
   for (int i = 0; i < dataSize; i++){
      cout<<data[i]<<" ";
   }
}

int median (int data[], int dataSize){
   insertionSort(data, dataSize);

   cout<<"\nThe median is"<<endl;

   if (dataSize % 2 != 0)
      return (int)data[dataSize / 2];

   return (int)(data[(dataSize - 1) / 2] + data[dataSize / 2]) / 2;

}

int main() {
   int dataSize, data[10];

   cout<<"Input data size "; cin>>dataSize;
```

```
    cout<<"Input element of array ("<<dataSize<<" much) ";
    for (int i = 0; i < dataSize; i++){
        cin>>data[i];
    }
    cout<<"The sorted data of ";
    for (int i = 0; i < dataSize; i++){
        cout<<data[i]<<" ";
    }
    cout<<"\nis "<<median(data, dataSize)<<endl;

    return 0;
}
```

Program:

```
Input data size 5
Input element of array (5 much) 5 7 6 2 1
The sorted data of 5 7 6 2 1
is 1 2 5 6 7
The median is
5
Program ended with exit code: 0
```

All Output ⌃ Filter

2. (40 points) Create a program allowing the user to choose:
    a. **Enter the number of data to be sorted**
    b. **Generate random data**
    c. Sort the data in ascending using the insertion sort method
    d. Sort the data in descending using the insertion sort method
    e. Sort the data in ascending using the selection sort method
    f. Sort the data in descending using the selection sort method

    Try with 100 data, 1000 data, 10000 data, and 100000 data. For each sort method, write the running time. Write the result in a table.

Source Code:

```
#include <iostream>

using namespace std;

void insertionAscend(int data[], int size){
    int i, temp;
```

```
    for(int j = 1; j < size; j++) { // array is going to be sort
        i = j-1;
        temp = data[j];
        while(data[i] > temp && i>=0){
            data[i+1] = data[i];
            i--;
        }
        data[i+1] = temp;
    }
    cout<<"The sorted data ascendingly using insertion sort is: "<<endl;
    for (int i = 0; i < size; i++){ //sorted array is going to be print
        cout<<data[i]<<" ";
    }
}

void insertionDescend(int data[], int size){
    int i, temp;
    for(int j = 1; j < size; j++) { // array is going to be sort
        i = j-1;
        temp = data[j];
        while(data[i] < temp && i>=0){
            data[i+1] = data[i];
            i--;
        }
        data[i+1] = temp;
    }
    cout<<"The sorted data descendingly using insertion sort is: "<<endl;
    for (int i = 0; i < size; i++){ //sorted array is going to be print
        cout<<data[i]<<" ";
    }
}

void selectionAscend(int data[], int size){
    int temp;
    for(int i = 0; i<size; i++) {
        int min = i;
        for(int j=1+i; j<size; j++){
            if(data[j] < data[min]){
            min = j;
        }
    }
    temp = data[i]; data[i] = data[min]; data[min] = temp;
    }
    for(int i = 0;i<size;i++){ cout<<data[i]<<" ";
    }
}
```

```
void selectionDescend(int data[], int size){
   int temp;
   for(int i = 0; i<size; i++) {
      int min = i;
      for(int j=1+i; j<size; j++){
         if(data[j] > data[min]){
         min = j;
      }
   }
   temp = data[i]; data[i] = data[min]; data[min] = temp;
   }
   for(int i = 0;i<size;i++){ cout<<data[i]<<" ";
   }
}

int main(){
   int size;
   cout<<"Input size of array that want to be generated "; cin>>size;

   //Generate random array with number between 0 to 99
   int random[size];
     for(int i=0;i<size;i++)
       random[i]=rand()%100;

     cout<<"\nElements of the array:"<<endl;

   for(int i=0;i<size;i++){
      cout<<random[i]<<" ";
   }
   cout<<endl;
   insertionAscend(random, size);
   insertionDescend(random, size);
   selectionAscend(random, size);
   selectionDescend(random, size);
   return 0;
}
```

Program: using 100 as input for array size

```
Input size of array that want to be generated 100

Elements of the array:
7 49 73 58 30 72 44 78 23 9 40 65 92 42 87 3 27 29 40 12 3 69 9 57 60 33 99 78 16 35 97 26 12 67 10 33 79 49 79 21 67 72 93 36 85 45 28 91 94 57 1 53 8 44 68 90 24 96 30 3 22 66 49 24 1 53 77 8 28 33 98 81 35 13 65 14 63 36 25 69
   15 94 29 1 17 95 5 4 51 98 88 23 5 82 52 66 16 37 38 44

The sorted data ascendingly using insertion sort is:
1 1 1 3 3 3 4 5 5 7 8 8 9 9 10 12 12 13 14 15 16 16 17 21 22 23 23 24 24 25 26 27 28 28 29 29 30 30 33 33 33 35 35 36 36 37 38 40 40 42 44 44 44 45 49 49 49 51 52 53 53 57 57 58 60 63 65 65 66 66 67 67 68 69 69 72 72 73 77 78 78
   79 79 81 82 85 87 88 90 91 92 93 94 94 95 96 97 98 98 99
The sorted data descendingly using insertion sort is:
99 98 98 97 96 95 94 94 93 92 91 90 88 87 85 82 81 79 79 78 78 77 73 72 72 69 69 68 67 67 66 66 65 65 63 60 58 57 57 53 53 52 51 49 49 49 45 44 44 44 42 40 40 38 37 36 36 35 35 33 33 33 30 30 29 29 28 28 27 26 25 24 24 23 23 22 21
   17 16 16 15 14 13 12 12 10 9 9 8 8 7 5 5 4 3 3 3 1 1 1
The sorted data ascendingly using selection sort is:
1 1 1 3 3 3 4 5 5 7 8 8 9 9 10 12 12 13 14 15 16 16 17 21 22 23 23 24 24 25 26 27 28 28 29 29 30 30 33 33 33 35 35 36 36 37 38 40 40 42 44 44 44 45 49 49 49 51 52 53 53 57 57 58 60 63 65 65 66 66 67 67 68 69 69 72 72 73 77 78 78
   79 79 81 82 85 87 88 90 91 92 93 94 94 95 96 97 98 98 99
The sorted data descendingly using selection sort is:
99 98 98 97 96 95 94 94 93 92 91 90 88 87 85 82 81 79 79 78 78 77 73 72 72 69 69 68 67 67 66 66 65 65 63 60 58 57 57 53 53 52 51 49 49 49 45 44 44 44 42 40 40 38 37 36 36 35 35 33 33 33 30 30 29 29 28 28 27 26 25 24 24 23 23 22 21
   17 16 16 15 14 13 12 12 10 9 9 8 8 7 5 5 4 3 3 3 1 1 1 Program ended with exit code: 0
```

3. (40 points) Create a program to implement the STL C++ sort() function. Try with 100 data, 1000 data, 10000 data, and 100000 data. Compare the running time with the result of the 1st problem.

Source code:

```
#include <iostream>

using namespace std;

int main(){
   int size;
   cout<<"Input size of array that want to be generated "; cin>>size;

   //Generate random array with number between 0 to 99
   int random[size];
     for(int i=0;i<size;i++)
       random[i]=rand()%100;

     cout<<"\nElements of the array:"<<endl;

   for(int i=0;i<size;i++){
      cout<<random[i]<<" ";
   }
   cout<<endl;
   sort(random, random + size);
   return 0;
}
```

Program:

```
Input size of array that want to be generated 100

Elements of the array:
7 49 73 58 30 72 44 78 23 9 40 65 92 42 87 3 27 29 40 12 3 69 9 57 60 33 99 78 16 35 97 26 12 67 10 33 79 49 79 21 67 72 93 36 85 45 28
    91 94 57 1 53 8 44 68 90 24 96 30 3 22 66 49 24 1 53 77 8 28 33 98 81 35 13 65 14 63 36 25 69 15 94 29 1 17 95 5 4 51 98 88 23 5 82
    52 66 16 37 38 44
1 1 1 3 3 3 4 5 5 7 8 8 9 9 10 12 12 13 14 15 16 16 17 21 22 23 23 24 24 25 26 27 28 28 29 29 30 30 33 33 33 35 35 36 36 37 38 40 40 42
    44 44 44 45 49 49 49 51 52 53 53 57 57 58 60 63 65 65 66 66 67 67 68 69 69 72 72 73 77 78 78 79 79 81 82 85 87 88 90 91 92 93 94 94
    95 96 97 98 98 99 Program ended with exit code: 0
```

Build time for 100 data using sort():