

ESE 2025 - Week 9 Report

Instructor: Takis Zourntos

Student: Vy Nguyen

Introduction:

This report shows the algorithm of finding a maximum, minimum, median integer in a vector. As well as using the built-in function to solve those requirements, and sorting an array in C++ program

Discussion:

To find a maximum value in a vector manually, we follow this pseudocode

```
declare variable i =1;
declare variable next, max;
assign max = first element ;
while i not equal to size of array
    assign next = i element
    if max < next
        max = next;
    increment of i;
return max
```

```
int Max(vector<int> thearr)
{
    int max = thearr[0];
    int next;
    vector<int>::size_type i =1;
    while (i != thearr.size())
    {
        next = thearr[i];
        if (max < next)
            max = next;
        ++i;
    }
    return max;
}
```

To find the minimum integer in the vector manually, we follow:

```
declare variable i =1;
declare variable next, min;
assign min = first element ;
while i not equal to size of array
    assign next = i element
    if min > next
        min = next;
    increment of i;
return min;
```

```
int Min(vector<int> thearr)
{
    int min = thearr[0];
```

```

int next;
vector<int>::size_type i =1;
while (i != thearr.size())
{
    next = thearr[i];
    if (min > next)
        min = next;
    ++i;
}
return min;
}

```

To find the median of the vector, we follow:

declare float median;

if size of array is even

median = [element(size / 2) + element(size / 2 - 1)] / 2;

else if size of array is odd

median = element(size/2);

return median;

C++

```

float Median(vector<int> thearr)
{
    float Median;
    if ((thearr.size() % 2) == 0)
        Median =
        (thearr[thearr.size()/2]+thearr[(thearr.size()/2)-1])/2.0f;
    else
        Median = thearr[(thearr.size()/2)];
    return Median;
}

```

Using the built-in function:

To use the functions, we need to use the algorithm library by `#include <algorithm>`

```

/* Maximum integer */
cout << "The maximum integer is " <<
*std::max_element(myarr.begin(),myarr.end())<< endl;
/* Minimum integer */
cout << "The minimum integer is " <<
*std::min_element(myarr.begin(),myarr.end())<< endl;

```

For sorting an array, we can use the built - in function with the same library algorithm

```

sort(myarr.begin(),myarr.end());

```

Summary:

Although we can write our own function for specific usage, some of the built-in functions are extremely helpful in larger programs.

Appendix:

```

//=====
// Name      : MaxMinAve.cpp
// Author    : Vy
// Version   :
// Copyright  : Your copyright notice
// Description : , Ansi-style
//=====

#include <iostream>
#include <vector>
#include <algorithm> // this library contain min_element and max_element built-in

using std::vector;
using std::cout;
using std::endl;
using std::cin;

/*
 *
 */
int Max(vector<int> thearr)
{
    int max = thearr[0];
    int next;
    vector<int>::size_type i =1;
    while (i != thearr.size())
    {
        next = thearr[i];
        if (max < next)
            max = next;
        ++i;
    }
    return max;
}
int Min(vector<int> thearr)
{
    int min = thearr[0];
    int next;
    vector<int>::size_type i =1;
    while (i != thearr.size())
    {
        next = thearr[i];
        if (min > next)
            min = next;
        ++i;
    }
    return min;
}
float Median(vector<int> thearr)
{
    float Median;
    if ((thearr.size() % 2) == 0)

```

```

        Median = (thearr[thearr.size()/2]+thearr[(thearr.size()/2)-1])/2.0f;
    else
        Median = thearr[(thearr.size()/2)];
    return Median;
}
int main()
{
    /* store integers in a vector from standard input */
    vector<int> myarr;                // our container
    int token;                       // our container content variable
    cout << "Please enter some integers, followed by <CTRL><D>:" << endl;
    while (cin >> token)             // loop exits when user enters <CTRL><D>
    {
        myarr.push_back(token);
    }
    cout << endl << endl;

    /* print integers from vector */
    cout << "Your integer array are:" << endl;
    for (vector<int>::size_type j = 0; j != myarr.size(); ++j)
    {
        cout << " " << myarr[j];
    }
    cout << endl;

    /* The largest integer */
    int max = Max(myarr);
    cout << endl << "The largest integer in the array is: " << max << endl;
    /*The smallest integer*/
    int min = Min(myarr);
    cout << endl << "The smallest integer in the array is: " << min << endl;
    /* The Median integer*/
    float median = Median(myarr);
    cout << endl << "The median of the array is: " << median << endl;
    /*
    * Maximum, Minimum, Median using built - in functions
    */
    cout << "Using built in functions"<< endl << endl;
    /* Maximum integer */
    cout << "The maximum integer is " << *std::max_element(myarr.begin(),myarr.end())<< endl;
    /* Minimum integer */
    cout << "The minimum integer is " << *std::min_element(myarr.begin(),myarr.end())<< endl;
    /*
    * Built-in functions for sorting an array
    */
    sort(myarr.begin(),myarr.end());
    cout << "The sorted array is"<<endl;
    for (vector<int>::size_type j = 0; j != myarr.size(); ++j)
    {
        cout << " " << myarr[j];
    }
    cout << endl;
}

```

```
/* exit happily */  
return 0;  
}
```