**Fundamentals of Programming**

**Lab Journal - Lab # 12**

Name: Mohammad Arsalan Shakil

Enrollment #: 01-134181-032

Class: BS(CS)-1A

**Objective**

This lab will cover

* basic pointer handling
* array notations
* passing arrays to functions
* calling functions within functions
* Revision of sorting and searching.

**Exercise 1**

Give answers to the following in one line only. Do NOT write the whole program.

|  |  |
| --- | --- |
| 1. | Declare an integer and a float and display their addresses in memory.  Int a;  Float b;  Int \*p,\*d;  p=&a;  d=&b; |
| 2. | Declare an integer and a pointer to integer. Assign the address of integer to the pointer. Display the address of the integer using pointer variable.  int a=10;  int \*p =&a;  cout<<p; |
| 3. | Display the value of the variable in the above question using the pointer.  cout << \*p; |
| 4. | Declare an integer and a pointer to integer. Assign the address of integer to the pointer. Assign a value 5 to the variable using the pointer.  Int x;  Int \*px;  px=&x;  \*px=5; |

**Exercise 2**

Write the output of the following code fragments.

|  |  |
| --- | --- |
| 1. | int x=10;  int \*px;  px= &x;  cout << px <<endl;  cout << &x <<endl; |
| Output: | |
| 2. | int x=10;  int \*px;  px= &x;  cout << \*px <<endl;  cout << x <<endl; |
| Output: | |
| 3. | int x=10;  int \*px;  px= &x;  \*px = 100;  cout << x <<endl; |
| Output: | |
| 4. | int firstvalue, secondvalue;  int \* mypointer;  mypointer = &firstvalue;  \*mypointer = 10;  mypointer = &secondvalue;  \*mypointer = 20;  cout << "firstvalue is " << firstvalue << endl;  cout << "secondvalue is " << secondvalue << endl |
| Output: | |

**Exercise 3 :**

Write the two ways of displaying the 4th element of an array **num** of type **float** and size **10.**

|  |  |
| --- | --- |
| **using array subscript notation:** | Cout<<num[3]; |
| **using array offset notation:** | Cout<<\*(num+3); |

**Exercise 4**

Write a C++ Program that sorts an array through bubble sort by swapping values and then finding median of that sorted array. Most importantly write your code in the following parts.

1. Make a function **Swap( )** that takes address of the two integers variables as arguments and swap those two variables.
2. Make a function **BubbleSort( )** which sorts an integer type array into ascending order. The function BubbleSort( ) takes the array and its size as argument.

It uses the above function swap while sorting to swap two array elements by passing those array elements to swap function.

1. Make a function **Median()** that takes an array and its size as arguments. Then it calls the above function BubbleSort() to sort the array. After sorting it will find the mean of that sorted array through following formual.
2. Write main program. Take the values of array in it from user. Call only the function named median( ) from main and pass it the array and its size.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 |

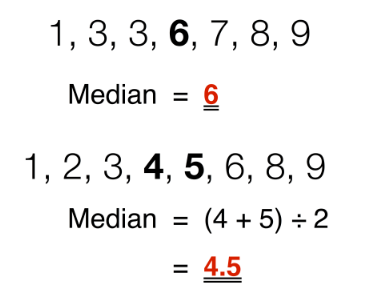
**Formula for Median of an array in C++**

**When size of array is even:** Median = arr[size/2];

**When size of array is odd:** Median = ( arr[ size/2 ]+arr[ size/(2+1)] ) / 2.0;

**Remember arrays index start from 0.**

**Example of median:**



**Code:**

#include<iostream>

#include<time.h>

#include<iomanip>

#include<ctype.h>

#include<cstring>

#include<string>

#include<math.h>

using namespace std;

double median(int arr[], int a);

void bubbleSort(int arr[], int n);

void swap(int \*x, int \*y);

int main()

{

const int size = 7;

int arr1[size];

cout << "Enter values for array to find median \n";

for (int i = 0; i < size; i++)

{

cout << "Enter " << i + 1 << " value : ";

cin >> arr1[i];

cout << endl;

}

float med = median(arr1, size);

cout << "Median Value is : " << med << endl;

system("pause");

}

double median(int arr[], int a)

{

int size = a;

double med;

bubbleSort(arr, a);

if (a % 2 != 0)

{

med = arr[size / 2];

}

else

{

med = (arr[size / 2] + arr[size / (2 + 1)]) / 2.0;

}

return med;

}

void bubbleSort(int arr[], int n)

{

int i, j;

bool swapped;

for (i = 0; i < n - 1; i++)

{

swapped = false;

for (j = 0; j < n - i - 1; j++)

{

if (arr[j] > arr[j + 1])

{

swap(&arr[j], &arr[j + 1]);

swapped = true;

}

}

if (swapped == false)

break;

}

}

void swap(int \*x, int \*y)

{

int temp = \*x;

\*x = \*y;

\*y = temp;

}

**Output:**

