Lab 10 Pointers and Call by Reference

Please test the correctness of your program in **Q-3** using **PASS**.

Q-1.

Write a program that asks the user to enter integers as inputs to be stored in the variables **A** and **B** respectively. There are also two integer pointers named **ptrA** and **ptrB**. Assign the addresses of **A** and **B** to **ptrA** and **ptrB** respectively and display the values in **A** and **B** using **ptrA** and **ptrB**.

Note:

1) Outputs may be different in different runs as we print addresses in program, the same hereinafter.
2) To manipulate the stream to print foo in hexadecimal use the hex manipulator:

```
cout << hex << foo;
```

Expected Outputs:

```
Enter value of A: 10
Enter value of B: 20
Value of ptrA is 10 stored in address 002EF83C
Value of ptrB is 20 stored in address 002EF830
```

Q-2.

Write a C++ program to find the max of an integer array. The program will ask the user to input the size of the array and the value of each element. The program prints on screen the maximum value in this array and the pointer that points to the maximum value.

Expected Outputs.

```
Enter number of values: 3
Enter values in array:
1 2 3
Largest integer value in the array is 3
Largest integer value in the array is stored in address 00659AC8
```

Q-3.

Given any two non-zero values from user, both can represent voltage (**V**) or current (**I**) and if one is **V**, the other will be **I**. Download the **resistance.cpp**, and modify this program to compute the possible values of resistance(**R**) from the user input, where **R=V/I**. For example, when the two inputted values are 5.1 and 2, **R** can be 5.1/2 = 2.55 or 2/5.1 = 0.39.

The program should be made up of the following four functions:

1) `getInput()`: get two values from user using **call by reference**, where the first one is **V** and the second is **I**. The return type of this function is void.
2) `toResistance()`: calculate the resistance **R** given **V** and **I**. The function should return a real number for the value of **R**.
3) `swap()`: swap the values of **V** and **I** to obtain the other possible pair of voltage and current using **call by reference.**
4) `main()`: call `getInput()` to obtain **V** and **I** using **call by reference**. After that, the program should pass **V** and **I** to the function `toResistance()` to obtain **R**. Then, swap the values of **V** and **I** using **call by reference** and pass them to `toResistance()` to obtain the other **R.** Finally, the possible values of **R** are printed (to 2 decimal places, with `cout << fixed << setprecision(2) << I << endl;`).

Expected output:

Example
Please enter two values: **5.1** **2** The value of one resistance is **2.55** The value of the other resistance is **0.39**

Q-4.

Download **ex1.cpp** and **ex2.cpp**. Compile and execute the program. Explain the output.

ex1.cpp

```
#include <iostream>
using namespace std;

int main() {
    int v = 5, *ptr;
    ptr = &v;
    *ptr = 42;
    cout << "v = " << v << endl;
    cin >> *ptr; // Let's enter 100
    // What happens if you write cin >> ptr; ?
    cout << "v = " << v << endl;
    v = 7;
    cout << "*ptr is " << *ptr << endl;
    cout << "Address of v is " << ptr << endl;
    return 0;
}
```

ex2.cpp

```
#include <iostream>
using namespace std;

void f(int *a, int *b) {
    int *c;
    c = a;
    *c = *c + 10;
    *b = *b + 10;
}

int main() {
    int x = 3, y = 4;
    int *ptr1;
    ptr1 = &x;
    f(ptr1, &y);
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "*ptr1 = " << *ptr1 << endl;
    return 0;
}
```

Please test the correctness of your programs using **PASS**.