TP 2 : Local copies and references, classes, STL

1 Reference and value of variables

1-1) Create a function "add2" declared as "void add2(int)" that adds 2 to the value of its input variable.

1-2) In a function "main", declare a variable "z" of type "int". Affect value "3" to this variable. Display variable "z", then apply function "add2", then display "z" once again. Did the value of "z" change ?

1-3) Modify the definition of function "add2", just replacing "int" by "int&" in its declaration. Use this function in "main" in the same manner as above. This times, did the value of "z" change ?

1-4) Before executing it, find out what will be the actual output of the program below.

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

void add1(int  x){x = x+1;}
void add2(int& x){x = x+2;}
void add3(int* x){*x = *x+2;}

int main(){
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  int* p; p = new int[4];
  p[0]=0; p[1]=1; p[2]=2; p[3]=3;
  cout << p[0] << "\t" << p[1] << "\t";
  cout << p[2] << "\t" << p[3] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  add1(p[2]);
  cout << p[0] << "\t" << p[1] << "\t";
  cout << p[2] << "\t" << p[3] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
```

```
  int*& s = p;
  add2(s[1]);
  cout << p[0] << "\t" << p[1] << "\t";
  cout << p[2] << "\t" << p[3] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  int** t = &s;
  (*t)++; (**t)++;
  cout << p[0] << "\t" << p[1] << "\t";
  cout << p[2] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  (*t)++; s = (*t)+1;
  cout << p[0] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  p = s-2;
  cout << s[0] << "\t" << s[1] << "\t";
  cout << s[2] << endl;
  cout << endl;
  //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%//
  s = s-1; add3(*t+3);
  cout << p[0] << "\t" << p[1] << "\t";
  cout << p[2] << "\t" << p[3] << endl;
  cout << endl;
}
```

2 Classes

Write a class representing "points" with two real coordinates, and test it in a main program. You have to equip this class with the following features :

i) default constructor
ii) a copy constructor
iii) a constructor that instantiates a point from its 2 real coordinates
iv) a method displaying the coordinates of the point
v) two methods allowing to set the coordinates of the point

3 Standard template library

3-1) Using the STL class `vector<T>` only, and no ordinary table, write a program that comply with the following features :

- the user is asked to type 10 integers
- these integers are stored in a vector
- the program sorts the integer values by increasing order
- the sorted sequence is printed in the terminal

3-2) Using the STL class `map<T,U>`, write a program that asks first to enter a series of pairs `(name,phone_number)`, as if the user were to fill a phone book. Then the program prints the whole content of the phone book in the terminal. In your program, it should be possible that the names be arbitrary words of variable length.