Object Oriented Programming with C++
Labwork 1 – *First steps with C++*

Exercise 1:

Which errors are found by C++ compiler with the following source file, but not with C compiler?

```
int main () {
  int a=10;
  int b=20;
  int c;
  c = g(a,b);
  printf ("value␣of␣g(%d,%d)=%d\n", a, b, c);
  return 0;
}

g (int x, int y) {
  return x*x + 2*y*y + y*y ;
}
```

Try it on computer using `gcc` and `g++`, with option `-pedantic`.

Exercise 2:

Translate in correct C the following C++ program:

```
#include <iostream>
using namespace std;
const int nb = 10;
const int excluded = 5;
int main () {
  cout<<"enter␣"<<nb<<"␣values:"<<endl;
  int values[nb];
  for (int i=0; i<nb; i++)
    cin >> values[i];
  int nbval = 0;
  for (int i=0; i<nb; i++)
    switch (values[i]) {
    case excluded-1:
    case excluded :
    case excluded+1:
      ++nbval;
    }
  cout<<nbval<<"␣forbidden␣values"<<endl;
  return 0;
}
```

Exercise 3:

Firstly, you have to find on Internet some documentation about the image file format PPM. I gave you a first function to read an existing image, and a second one to convert a colored image to a black-and-white image.

You have to complete the following third function in (very simple) C++:

1. `void write_ppm(Image img, const char*const filename);` that writes to a given file a black-and-white image in PPM format (using `P1` format).

Then, you have to test your functions in a program that:

- reads an image whose name is given in command line;
- converts this image into black-and-white using a simple threshold operation (all pixels below the threshold value are black, all above are white), the threshold value being given in command line;
- and then write the resulting image into a new image file whose name is given in command line.

Obviously, you have to use C++ specific input/output.

Exercise 4: The same thing

Transform the previous exercise using a class Image. The function that creates the image is a "constructor", the function for writing a classical member function.