Programming Exam 16/06/2021

Exercise 1

Create files `pcx.h` and `pcx.cpp` that allow to use the following function declaration:

```
bool load_pcx(const std::string& filename, mat<uint8_t>& img);
```

The definition of template `class mat<T>` is given in the attached `mat.h` file.

The function should open the **1 bpp PCX** file specified by `filename`, read its data and store it in the 8 bit per pixel matrix. Notice that every (decompressed) byte of the PCX image will produce 8 pixels in `img`. If a bit is 0, the corresponding pixel shall have value 0, if a bit is 1, the corresponding pixel shall have value 255.

Exercise 2

Create files `pcx.h` and `pcx.cpp` that allow to use the following function declaration:

```
bool load_pcx(const std::string& filename, mat<vec3b>& img);
```

The definition of template `class mat<T>` is given in the attached `mat.h` file. The declaration of type vec3b is provaided in the attached `types.h` file and is:

```
using vec3b = std::array<uint8_t, 3>;
```

The function should open the **24 bpp PCX** file specified by `filename`, read its data and store it in the `vec3b` matrix. Notice that colors are stored by plane in each scanline.

Exercise 3

Create files `pcx.h` and `pcx.cpp` that allow to use the following function declaration:

```
bool load_pcx(const std::string& filename, mat<vec3b>& img);
```

The definition of template `class mat<T>` is given in the attached `mat.h` file. The declaration of type vec3b is provaided in the attached `types.h` file and is:

```
using vec3b = std::array<uint8_t, 3>;
```

The function should open the **256 indexed color PCX** file specified by `filename`, read its data and store it in the `vec3b` matrix. Notice that colors are stored as 8 bpp indexes in the palette which is appended at the end of the compressed image data.