PPM-Load

Submit Solution

Una software house sta realizzando una app per dispositivi mobili che deve trasmettere dati in formato JSON. Uno dei campi richiesti è però una piccola immagine a colori RGB e JSON non supporta la trasmissione di dati binari. Per questo motivo la software house decide di utilizzare un semplice encoding Base64. L'immagine viene compressa a piani separati con l'algoritmo PackBits e poi ogni piano così compresso viene codificato in Base64. Le informazioni vengono passate come campi di un dizionario JSON. Ad esempio:

```
{
    "width": 6,
    "height": 6,
    "red": "3f+A",
    "green": "/gD+//4A/v/+AP7//gD+//4A/v/+AP7/gA==",
    "blue": "3QCA"
}
```

In questo esempio è codificata un'immagine con tre colonne rosse (255,0,0) e tre gialle (255,255,0), come in figura:

r	r	r	g	g	g
r	r	r	g	g	g
r	r	r	g	g	g
r	r	r	g	g	g
r	r	r	g	g	g
r	r	r	g	g	g

quindi i piani colore contengono (i valori sono mostrati qui in esadecimale e c'è uno spazio più grande tra una riga e l'altra, solo per chiarezza):

```
R:    FF,FF,FF,FF,FF,FF,    FF,FF,FF,FF,FF,FF,    FF,FF,FF,FF,FF,FF,
    FF,FF,FF,FF,FF,FF,    FF,FF,FF,FF,FF,FF,    FF,FF,FF,FF,FF,FF

G:    00,00,00,FF,FF,FF,    00,00,00,FF,FF,FF,    00,00,00,FF,FF,FF,
    00,00,00,FF,FF,FF,    00,00,00,FF,FF,FF,    00,00,00,FF,FF,FF

B:    00,00,00,00,00,00,    00,00,00,00,00,00,    00,00,00,00,00,00,
    00,00,00,00,00,00,    00,00,00,00,00,00,    00,00,00,00,00,00
```

Dopo la codifica PackBits:

```
R:    DD,FF,80
G:    FE,00,FE,FF,FE,00,FE,FF,FE,00,FE,FF,FE,00,FE,FF,FE,00,FE,FF,FE,00,FE,FF,80
B:    DD,00,80
```

Dopo l'encoding in Base64:

```
R:    3f+A
G:    /gD+//4A/v/+AP7//gD+//4A/v/+AP7/gA==
B:    3QCA
```

Nel file `process_ppm.cpp` implementare la funzione corrispondente alla seguente dichiarazione:

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

La funzione deve caricare un'immagine a colori nel formato PPM, seguendo le specifiche allegate. Se la lettura va a buon fine la funzione ritorna `true`, altrimenti ritorna `false`. L'implementazione della classe `mat` e della classe `vec3b` sono disponibili nei file `mat.h`, `ppm.h` e `ppm.cpp`.

Sono dati a titolo di esempio i file `test.ppm` e `facolta.ppm`.