

Tema 3. Gigel si captcha-urile



Structura formatului BMP

Vom lucra cu fisiere **BMP**, deci, cu **fisiere binare**.

O imagine BMP are urmatoarea structura:

- un **File Header** care are urmatoarele campuri:
 1. **signature** – 2 octeti - literele 'B' si 'M' in ASCII;
 2. **file size** – 4 octeti – dimensiunea intregului fisier;
 3. **reserved** – 4 octeti – nefolosit;
 4. **offset** – 4 octeti – offsetul de la inceputul fisierului pana la inceputului bitmap-ului, adica al matricii de pixeli.
- un **Info Header** care poate avea structuri diferite, insa noi vom lucra cu cel care se numeste **BITMAPINFOHEADER**. Are urmatoarele campuri:
 1. **size** – 4 octeti – dimensiunea Info Header-ului, care are o valoare fixa, 40;
 2. **width** – 4 octeti – latimea matricii de pixeli (numarul de coloane);
 3. **height** – 4 octeti – inaltimea matricii de pixeli (numarul de randuri);
 4. **planes** – 2 octeti – setat la valoarea fixa 1;
 5. **bit count** – 2 octeti – numarul de biti per pixel. In cazul nostru va avea mereu valoarea 24, adica reprezentam fiecare pixel pe 3 octeti, adica cele 3 canale, RGB;
 6. **compression** – 4 octeti – tipul de compresie. Acest camp va fi 0;
 7. **image size** – 4 octeti – se refera la dimensiunea matricii de pixeli, inclusiv padding-ul adaugat (vedeti mai jos);
 8. **x pixels per meter** – 4 octeti – se refera la rezolutia de printare. Pentru a simplifica putin tema, veti seta acest camp pe 0. Nu o sa printam imaginile :).
 9. **y pixels per meter** – la fel ca mai sus;
 10. **colors used** – numarul de culori din paleta de culori. Aceasta este o sectiune care va lipsi din imaginile noastre BMP, deoarece ea se afla in general imediat dupa **Info Header** insa doar pentru imaginile care au campul **bit count** mai mic sau egal cu 8. Prin urmare, campul va fi setat pe 0;
 11. **colors important** – numarul de culori importante. Va fi, de asemenea, setat pe 0, ceea ce inseamna ca toate culorile sunt importante.
- **BitMap**-ul, care este matricea de pixeli si care ocupa cea mai mare parte din fisier. Trei lucruri trebuie mentionate despre aceasta:
 1. pixelii propriu-zisi se afla intr-o matrice de dimensiune **height x width**, insa ea poate avea o dimensiune mai mare de atat din cauza **paddingului**. Acest padding este adaugat la sfarsitul fiecărei linii astfel incat fiecare linie sa inceapa de la o adresa (offset fata de inceputul fisierului) multiplu de 4. Mare atentie la citire, pentru ca acest padding trebuie **ignorat** (fseek). De asemenea, la scriere va trebui sa puneti **explicit** valoarea 0 pe toti octetii de padding.
 2. este **rasturnata**, ceea ce inseamna ca prima linie in matrice contine de fapt pixelii din extremitatea de jos a imaginii. Vedeti exemplul de mai jos;
 3. canalele pentru fiecare pixel sunt in ordinea BGR (**B**lue **G**reen **R**ed).

Header-ele pe care le puteti folosi in implementare se afla in scheletul de cod asociat temei.

Urmatiti cu foarte mare atentie exemplul de [aici](#) si incercati sa intelegeti cum e reprezentata o imagine BMP **inainte** de a incepe implementarea. Daca e ceva neclar, puteti intreba oricand pe forum.

Imaginea pe care o primiti are fundalul perfect alb (255, 255, 255) si contine doar cifre scrise cu diferite nuante de culori. Cifrele au dimensiunea de 5×5 pixeli si nu se pot intersecta. Daca notam cu "X" un pixel colorat si cu "." un pixel alb cifrele arata in felul urmator:

```
XXXXX  ....X  XXXXX  XXXXX  X...X  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
X...X  ....X  ....X  ....X  X...X  X....  X....  ....X  X...X  X...X
X...X  ....X  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  ....X  XXXXX  XXXXX
X...X  ....X  X....  ....X  ....X  ....X  X...X  ....X  X...X  ....X
XXXXX  ....X  XXXXX  XXXXX  ....X  XXXXX  XXXXX  ....X  XXXXX  XXXXX
```

Cerinte

Toate taskurile (mai putin bonusul) vor procesa o singura imagine. Numele acesteia o sa fie citit de pe prima linie din fisierul de intrare (vezi formatul datelor de intrare). Scopul fiecarui task este crearea unui nou fisier (binar pentru taskurile 1 si 3 si text pentru taskul 2) dupa urmatoarele reguli:

- se elimina extensia ".bmp" din fisierul initial.
- se adauga:
 - "_task1.bmp" pentru primul task;
 - "_task2.txt" pentru cel de-al doilea task;
 - "_task3.bmp" pentru taskul trei.

Exemplu: daca imaginea initiala se numeste "**captcha.bmp**", rezolvarea completa a celor trei taskuri presupune existenta fisierelor "**captcha_task1.bmp**", "**captcha_task2.txt**" si "**captcha_task3.bmp**" dupa rularea programului.

Bonusul va procesa o imagine diferita (citita din acelasi fisier de intrare) si presupune formarea unui noi imagini dupa reguli asemanatoare primelor taskuri, adaugand insa "**_bonus.bmp**" la final de nume. (imaginea "**captchaB.bmp**" va crea fisierul text "**captchaB_bonus.bmp**").

Task1

Inainte de a incerca rezolvarea captcha-ului, Gigel va propune sa incepeti cu ceva simplu: schimbarea culorii tuturor cifrelor din imagine. Pe langa imaginea cu care o sa lucrati la primele 3 taskuri o sa mai primiti 3 numere intregi intre 0 si 255 (in formatul BGR) reprezentand noua culoare dorita a cifrelor. Rezolvarea corecta a taskului presupune existenta unei imagini noi cu aceleasi cifre, dar colorate dupa regula data.

Exemplu input:



(captcha.bmp)

+ valorile 0, 255, 0 (B, G, R)

Exemplu output:



(captcha_task1.bmp)

Observatii:

- cu exceptia matricei de pixeli, toate celelalte valori din imaginea initiala (valorile campurilor din headere) trebuie sa ramana neschimbate;

▪

Task2

Acum, dupa ce ati reusit (sau nu) sa colorati cifrele din captcha a venit momentul sa incercati si recunoasterea lor. Numarul format va avea cifrele ordonate de la stanga la dreapta.

Exemplu input:



(captcha.bmp)

Exemplu output:

26908109 (captcha_task2.txt)

Observatii:

- cifrele pot incepe oriunde in imagine (atata timp cat nu depasesc dimensiunea imaginii).
- cifrele nu se vor intersecta si oricare doua cifre vor fi separate de cel putin o linie sau o coloana;
- nu vor exista 2 cifre care sa inceapa pe aceeasi coloana, putand astfel sa spunem pentru oricare 2 cifre care se afla la stanga celeilalte;
- pentru simplitate, putem considera ca fiecare cifra reprezinta un patrat de pixeli de dimensiune 5 * 5. Deci, nu va aparea urmatorul caz in fisierele de test:

-
- `XXXXX...X.`
- `X...X...X.`
- `X...X...X.`
- `X...X...X.`
- `XXXXX...X.`
-

- Testul de mai sus nu e valid din cauza cifrei unu. Aceasta trebuie reprezentata ca o matrice de 5 * 5 pixeli in care primele 4 coloane sa fie albe (lucru care nu se intampla).

▪

Task3

Vazand ca lucurile merg bine si este pe cale sa devina milionar, Gigel mai are o rugaminte la voi. Doreste sa faca o statistica a cifrelor care apar si sa vada cum ar arata captcha-urile fara anumite cifre. O sa primiti o lista de cifre pe care va trebui sa le eliminati. Dupa eliminarea unei cifre toate celelalte cifrele ce se afla la dreapta ei vor fi mutate spre stanga in felul urmator:

- prima cifra de dupa o cifra eliminata va lua locul cifrei eliminate;
- urmatoarea cifra va lua locul cifrei recent mutate;
- se va repeta pasul 2 pana nu mai raman cifre;

Exemplu input:



(captcha.bmp)

+ vectorul de valori: "2 0 9"

Exemplu output:



(captcha_task3.bmp)

Observatii:

- dimensiunea imaginii precum si celelalte campuri din headerele imaginii originale vor ramane neschimbate si in cadrul acestui task;