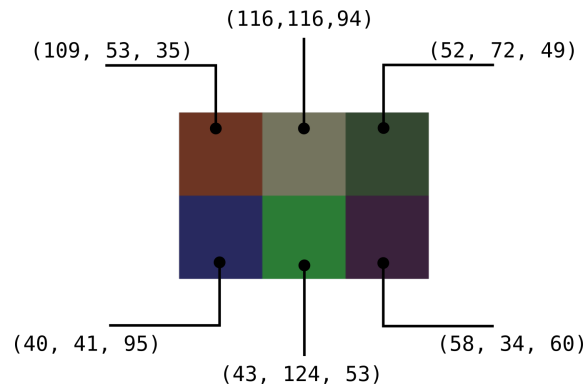


Семинар #10: Файлы. Работа с изображениями. Домашнее задание.



ppm - это простейший формат для хранения изображений. В отличие от других, более продвинутых форматов (например `jpg` или `png`), он не использует никаких хитрых алгоритмов сжатия, а просто хранит RGB значения каждого пикселя. Представленное выше крохотное изображение (3 на 2 пикселя) может быть представлено в формате `ppm` двумя эквивалентными способами - текстовым и бинарным. Цвета картинки специально подобраны так, чтобы в бинарном представлении получались печатаемые символы из таблицы ASCII (символы с кодами от 32 до 126). Так бывает не всегда. Если значение компоненты цвета выйдет за эти пределы, то текстовый редактор напечатает какой-нибудь непонятный символ (какой именно – зависит от кодировки) или вообще ничего не напечатает. Вот как будет выглядеть `ppm`-изображение выше, если открыть его в текстовом редакторе:

Текстовый формат

```
P3
3 2
255
109 53 35 116 116 94 52 72 49
40 41 95 43 124 53 58 34 60
```

Бинарный формат

```
P6
3 2
255
m5#tt^4H1()_|5:"<
```

Побайтовое представление файлов:

```
50 33 0a 33 20 32 0a 32 35 35 0a 31 30
39 20 35 33 20 33 35 20 0a 31 31 36 20
31 31 36 20 39 34 20 0a 35 32 20 37 32
20 34 39 0a 34 30 20 34 31 20 39 35 20
0a 34 33 20 31 32 34 20 35 33 20 0a 35
38 20 33 34 20 36 30
```

```
50 36 0a 33 20 32 0a 32 35 35 0a 6d 35
23 74 74 5e 34 48 31 28 29 5f 2b 7c 35
3a 22 3c
```

Для справки - коды ASCII нужных символов:

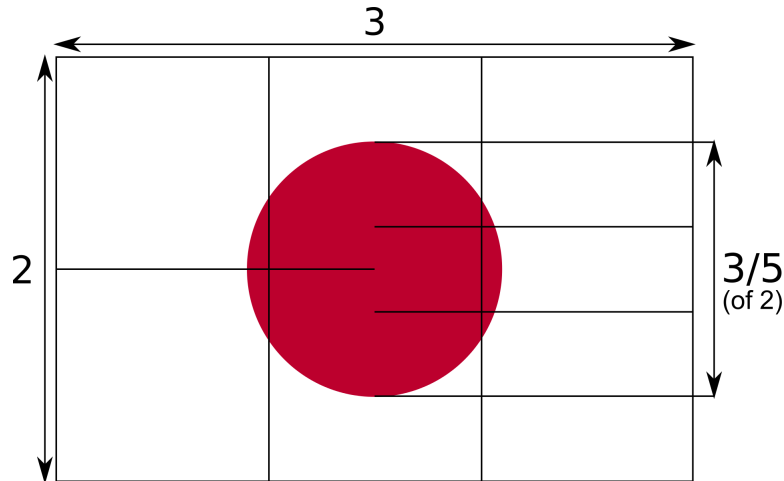
Символ	ASCII-код	ASCII-код в 16-ричной системе	Символ	ASCII-код	ASCII-код в 16-ричной системе
P	80	50	0	48	30
пробел	32	20	1	49	31
\n	10	0a	2	50	32
m	109	6d	3	51	33
#	35	23	4	52	34
t	116	74	5	53	35

В файле `tiny.c` представлена программа, в которой создаются 2 файла изображения. Скомпилируйте эту программу и запустите. Создадутся 2 файла: изображение данной маленькой картинки в текстовом и бинарном форматах.

В этих задачах вам понадобится программа для просмотра изображений в формате `ppm`. Если у вас нет программы, которая поддерживает этот формат на компьютере, то советую использовать IrfanView: www.irfanview.com. В этой программе по умолчанию используется сглаживание при приближении. Его можно отключить, чтобы было видно каждый пиксель View -> Display Options -> Use Resample for zooming (убрать галочку).

Рисование в файл изображения

- **Флаг:** В файле `flag.c` содержится пример работы создания изображения в формате `.ppm` в программе. Напишите программу, которая будет рисовать флаг Японии. Изображение должно иметь размер 600 на 400 пикселей. Компоненты белого цвета: (255, 255, 255). Компоненты красного цвета: (190, 0, 41).



- **Случайные круги:**
 - Напишите функцию `void draw_circle(Color* data, int width, int height, int x0, int y0, int r, Color c)` которая будет рисовать круг на холсте `data` с центром в точке `(x0, y0)`, радиусом `r` и цветом `c`.
 - Напишите программу, которая будет рисовать `n` кругов случайного цвета, расположения. Радиус тоже выбирается случайный в диапазоне от `a` до `b`. Параметры `n`, `a` и `b` передаются через аргументы командной строки. Программа должна создавать изображение `circles.ppm`.
- **Функция двух переменных:** Напишите программу, которая будет рисовать значения функции двух переменных $f(x, y)$ в области $[-1, 1] \times [-1, 1]$. Значения функции должны сохраняться в изображении размером 500 на 500 пикселей. К примеру, пиксель с координатами (250, 250) должен хранить значение функции в точке (0, 0), а пиксель (0, 400) - значение в точке (-1, 0.6). Учтите, что значения пикселей изображения должны лежать в интервале от 0 до 255. Постройте изображения следующих функций:

1. $f(x, y) = k \cdot |x \cdot y|$
2. $f(x, y) = k \cdot |\sin(10 \cdot (x^2 + y^2))|$
3. $f(x, y) = k \cdot |\sin(5000 \cdot (x^2 + y^2))|$
4. $f(x, y) = k \cdot |\cos(10x) \cdot \sin(10y)|$
5. $f(x, y) = k \cdot \frac{1}{2} \cdot \left| \sin\left(\frac{3}{0.1+|x|}\right) + \sin\left(\frac{3}{0.1+|y|}\right) \right|$

Параметр $k = 255$ подбирается так, чтобы значения компонент цвета лежало в диапазоне от 0 до 255.

Обработка изображений

В файле `brightness.c` содержится программа, которая увеличивает яркость изображения. Используйте её как пример для решения следующих задач. Компиляция и запуск этой программы осуществляется следующим образом:

```
gcc -std=c99 -o brighter brightness.c
./brighter images/emir.ppm 50
```

- **Черно-белое изображение:** Написать программу, которая принимает на вход файл изображения, считывает его и превращает в чёрно-белое изображение и записывает в файл `result.ppm`. Название изображения должно передаваться через аргументы командной строки.
- **Перестановка цветов:** Написать программу, которая переставляет местами красную и синюю компоненты цвета. Применить её на файле `emir.ppm`.
- **Сепия:** Написать программу, которая будет применять к изображению эффект сепии.

Формулы для эффекта сепии:

$$r_{new} = 0.393 \cdot r + 0.769 \cdot g + 0.189 \cdot b$$

$$g_{new} = 0.349 \cdot r + 0.686 \cdot g + 0.168 \cdot b$$

$$b_{new} = 0.272 \cdot r + 0.534 \cdot g + 0.131 \cdot b$$

Если какое-то из этих значений станет большим, чем 255, то его нужно приравнять к 255.



- **Отражение:** Напишите программу, которая зеркально отражает изображение по вертикали (относительно горизонтальной прямой).
- **Свёртка изображения. Размытие.** Операция свёртки изображения задаётся следующей формулой:

$$data_{new}[i, j] = \sum_{p=-1}^1 \sum_{q=-1}^1 K[p+1, q+1] \cdot data[i+p, j+q]$$

, где K - некоторая матрица 3 на 3. Для размытия эта матрица равна:

$$K = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Подробнее о свёртке можно посмотреть тут: www.youtube.com/watch?v=C_zFhWdM4ic

Если провести эту операцию 1 раз, то размытие будет небольшое. Чтобы размыть изображение сильнее нужно повторить эту операцию несколько раз. Напишите программу, которая будет размывать изображение n раз (n передаётся через аргументы командной строки).

- **Консольный графический редактор:** Объедините все решения предыдущих задач в одну программу `mge`. Выбор эффекта должен задаваться с помощью аргументов командной строки. Например так:

```
mge --sepia image.ppm result.ppm
```

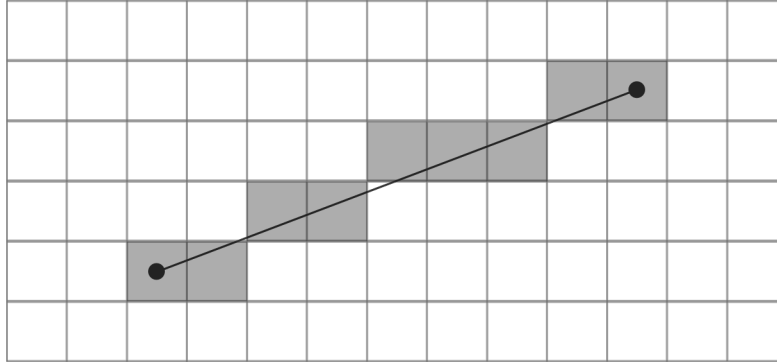
Программа должна применять эффект сепии на изображение `image.ppm` и сохранять результат в `result.ppm`. А при таком вызове программа должна применять эффект размытия 10 раз:

```
mge --blur 10 image.ppm result.ppm
```

Добавьте ещё опции: `--brighter`, `--bw`, `--change colors`, `--mirror`.

Рисование линий. Алгоритм Брезенхема:

Алгоритм построения прямой линии между двумя точками на двумерном холсте называется алгоритмом Брезенхема. В файле `4lines.c` есть реализация этого алгоритма на языке C.

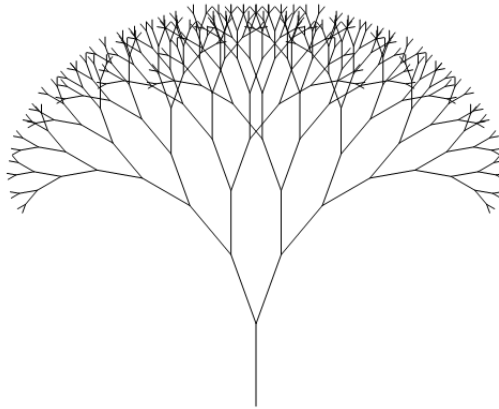


- **Случайные линии:**

Создать программу, которая будет рисовать n случайных отрезков случайного цвета. n должен передаваться через аргументы командной строки. Программа должна создавать файл `randlines.ppm`.

- **Фрактальное дерево:**

Напишите рекурсивную функцию, которая будет рисовать фрактальное дерево:



Работа с изображениями формата .jpg и .png:

Популярные форматы изображений такие как `.jpg` или `.png` имеют сложную структуру, так как используют хитрые алгоритмы сжатия. Про `jpg` можно посмотреть тут:

→ www.youtube.com/watch?v=8N0Bx8DMt6c

Считывать файлы таких форматов самостоятельно было бы очень непросто, так как нужно было бы досконально знать строение файла изображения. К счастью делать это необязательно, так как уже всё сделано за нас другими программистами. Есть простая библиотека `stb_image`. Эта библиотека подключается простым `include` (что не справедливо для большинства других библиотек). Пример работы с библиотекой – в папке `stb`.

- **jpeg редактор:** Измените программу из задачи консольный графический редактор так, чтобы она работала с форматом `.jpg`. Добавьте параметр командной строки `quality`, который будет задавать качество получаемой `jpg`-картинки.