Задачи за домашна работа № 2

1. От подаден от потребителя mp3 файл, да се прочете ID3v1 tag, който е записан в файла. Да се изведе неговото съдържание. Ако няма такъв, да се генерира нов, от името на файла, ако името му е подходящо. Генерирането разпознава името на артиста и песента по следния шаблон: "<artist> - <title>.mp3". Да се вземе впредвид ако потребителя е въвел пълен път до файла.

   Примерен вход:
   C:\Eagle Eye Cherry - save tonight.mp3

   Примерен изход:
   Artist: Eagle Eye Cherry
   Title: Save Tonight
   Year: 1998
   Genre: Alt. Rock

   Пояснения: в ID3v1 tag се запазва мета-информация за песента към музикалния файлов формат mp3. Тя се записва в последните 128 Byte-а на файловете, като винаги започва с буквите TAG. Повечето музикални програми, които могат да възпроизвеждат mp3 формат, могат да го покажат. Формата на информацията може да видите тук: http://en.wikipedia.org/wiki/ID3#ID3v1

2. Изкарва на екрана всички смислени низове от даден бинарен файл подаден от потребителя. За смислен низ се приема последователност от символи, не по-къса от 3 символа и съдържаща само английски букви, числа, разстояние, нов ред, табулация и препинателни знаци.

   Примерен вход:
   calc.exe

   Примерен изход:
   This program cannot be run in DOS mode.
   SHELL32.dll
   SHLWAPI.dll
   …

3. Изкарва на екрана честотата (колко пъти се срещат) на всички символи от въведен от потребителя низ.

   Примерен вход:
   Hello World

   Примерен изход:
   H – 1; e – 1; l – 3; o – 2; W – 1; r – 1; d – 1;

   Добавка: за да се усложни задачата, може да се укаже буквите да се броят case-insensetive.

4. Изкарва на екрана броя думи и броя изречения в даден файл зададен от потребителя.

   Примерен вход:
   SomeFile.txt

   Примерен изход:
   Words count: 53
   Sentences: 10

5. Да се прочете HTML файл и да изкара на екрана всички линкове към външни ресурси, които се зареждат динамично за следните тагове-атрибути: javascript - src, img - src, link - href (css).

   Примерен вход:
   index.html

   Примерен изход:
   http://code.jquery.com/jquery-1.9.0.js
   css/style.css
   images/background.png
   …

   Пояснение: HTML е описателен език, който указва на браузера как да рендерира дадена страница. Основен конструктивен елемент в този език е „tag”, който може да има 1 или повече атрибути. В горе-споменатите тагове, съответните атрибути служат да укажат на браузера да заредят динамично дадения ресурс от посочения адрес. Пример:
   **

6. Проверява дали подадения (бинарен) файл има правилен extension от набор поддържани extension-и (bmp, jpg, png, gif, exe, doc, mp3). Ако не е правилен, пита дали да преименова файла на правилния extension и го прави.

Примерен вход:	Примерен изход:
photo.bmp	The correct filename extension should be png. Should I correct the file name?

Пояснение: всеки тип файл обикновенно кодира в първите си няколко байтове своя тип, за да може правилно да бъде разпознат от дадена програма и да не разчита на разширението на файла, което може да бъде променено по погрешка от потребителя. Пример: всички png файлове започват с „‰PNG“ символи в първите 4 байта. Може да се види с hex editor. Повече информация може да се намери тук:
http://en.wikipedia.org/wiki/Magic_number_%28programming%29#Magic_numbers_in_files

7. Да се прочете число с научна нотация от string в double и да се изведе на екрана.

Примерен вход:	Примерен изход:
3.456789e+002	345.6789

8. Превръщане от арабски цели числа в римски цифри:
1-I; 5-V; 10-X; L-50; C-100; D-500; M-1000.

Примерен вход:	Примерен изход:
14	XIV

9. Превръщане от римски цифри в арабски числа:
1-I; 5-V; 10-X; L-50; C-100; D-500; M-1000.

Примерен вход:	Примерен изход:
XIV	14

10. Изписване на int число с букви (на български).

Примерен вход:	Примерен изход:
23	Dvadeset i tri

11. Табулатор на код – табулира (слага '\t' пред всеки ред) редовете спрямо дълбочината на вложените блокове, слага нов ред около всяка отваряща и затваряща скоба. Прочита посочен .c файл и го форматира.

Примерен входен файл:
```
int main() { if (1) {printf("blah\n");} return 0}
```

Примерен изходен файл:
```
int main()
{
   if (1)
   {
      printf("blah\n");
   }
   return 0;
}
```

12. Прочитане на колоните от *.csv файл в многомерен масив и извеждането му на екрана. Файла трябва да има следните колони: FakNo, Name, Family, Age, Mark; за разделител се използва запетайка и трябва да съдържа първи заглавен ред с имената на колоните. Да се провери дали заглавния ред съответства на изискванията за файла (case-insensetive). Да се изведе средно-аритметичната стойност на колона Mark (оценка).

Примерен входен файл:
FakNo, Name, Family, Age, Mark
121212099, Petar, Petrov, 20, 5.34
121212521, Ivan, Ivanov, 21, 3.45
121212111, Georgi, Georgiev, 20, 4.5

Примерен изход:
The average mark for the group is: 4,43

Пояснение: csv файлов формат представлява текстови файл изобразяващ таблица с данни. Първия ред обикновенно е заглавен и съдържа името на колоните. Следващите редове са самите данни за съответните колони. Колоните се отделят една от друга с избран символ, в нашия случай запетайка.

Добавка: за да се усложни задачата, може да се даде възможност на потребителя да изтрива редове/променя данни/смята други неща след което запазва промените отнов в файл.

13. Да се прочете даден *.sub файл с MicroDVD формат и да се изместят кадрите напред или назад с число подадено от потребител.

Примерен вход: cool_movie.sub -10

Примерен входен .sub файл:
{30}{50} Big ass title
{80}{120} Some actor name

Примерен изходен .sub файл:
{20}{40} Big ass title
{70}{110} Some actor name

*Пояснение: *.sub е разпространен файлов формат съдържа субтитри съпътстващи даден филмов файл. Понякога потребители попадат на субтитри, които просто са изместени напред или назад и не съвпадат с говора на актьорите. *.sub файла може да съдържа различен вид формат на данните. Избрания MicroDVD борави с брой кадри, докато другите с минути и секунди и т.н. За повече информация: http://en.wikipedia.org/wiki/MicroDVD*

14. Да се превърне *.sub файл с MicroDVD формат в *.srt файлов формат, т.е. от изминали кадри в време (секунди, минути, часове и т.н.). Да се даде възможност на потребителя да укаже скоростта на възпроизвеждане на филма т.е. на колко fps ще върви.

Примерен входен .sub файл:
{30}{50} Big ass title
{80}{120} Some actor name

Примерен изходен .srt файл:
1
00:00:01,500 --> 00:00:01,800
Big ass title

2
00:00:02,300 --> 00:00:02,400
Some actor name

*Пояснение: *.sub е разпространен файлов формат съдържа субтитри съпътстващи даден филмов файл. *.sub файла може да съдържа различен вид формат на данните. Избрания MicroDVD борави с брой кадри, докато другите с минути и секунди и т.н. За повече информация: http://en.wikipedia.org/wiki/MicroDVD*

**.srt формат е друг разпространен файл формат съдържащ субтитри. Неговия формат борави с времена в формат: hours:minutes:seconds,milliseconds: http://en.wikipedia.org/wiki/SubRip*

15. Да се кодира даден файл с шифър на Цезар в друг файл. Шифъра на Цезар представлява кодиране на всеки символ от изречението с друг, като към оригинала се прибавя избрано число - ключ, например 2. Ключа трябва да се въвежда от потребителя.

Примерен входен файл:	Примерен изходен файл с ключ 2:
ABC	DEF

16. Електронен четец: програма, която показва за четене кой да е текстови файл, на страници. При въвеждане 'n' или 'p', програмата изчиства целия екран, прочита следващата/предишната порция от дадения файл (примерно следващите 200 символи) и ги показва като страница от файла. 'q' символа е за quit.

Особености: нарочно съм подал 'n' и 'p' като навигационни букви, вместо стрелките, тъй като в C май няма стандартен начин да вземеш настиснат клавиш. getch() върши работа, но той е conio.h. Изчистването на екран също е проблем. Един вариант би бил с #ifdef да изпълнява system("cls"); за Windows и system("clear") за Linux ако не се лъжа.

Добавка: задачата може да се усложни с команди за find, да показва на коя от колко страници си и т.н.

17. Hex viewer: програма, която показва бинарно за четене кой да е файл, на страници. При въвеждане 'n' или 'p', програмата изчиства целия екран, прочита следващата/предишната порция от дадения файл (примерно следващите 50 байта) и ги показва като страница от файла. 'q' символа е за quit. Байтовете се показват в шестнайсетичен вид в 16 колони в средата. В дясно, на същия ред срещу тях да се показват техните съответни символи. За нестандартни символи, които не са видими, може да се използва служебен символ точка.

Примерен изход:

FF FF FF FF E6 20 00 00 00 EE F2 F2 79 65 3C 99	яяяяж ... оттуе<™
71 04 A1 95 79 A1 99 7D 69 55 20 EE CE 18 FA DA	q□Ў•уЎ™}iU оО□ъЪ

Особености: същите забележки от горната задача важат и за тази.

18. Да се реализира морски шах с размер 3x3. Да се изрисува игралното поле с ASCII символи (включително и маркираните полета с 'X' и 'O' символи). Да се редуват двама играчи, които въвеждат числата от 0 до 9, като числата от numpad блока на клавиетурата съответстват на възможните полета в играта. При всеки въведен ход, екрана се прерисува наново с попълнените полета.

Особености: подобно на горните задачи, прерисуването на екран е проблем. Местенето на курсор също не е стандартна операция и изисква допълнителни dependency-та. Затова според мен най-добрия избор си остава system("...");

19. Да се реализира програма, която при въведен файл и въведен максимален размер limit, разцепва файла на по-малки файлове с размер limit. Имената на новите файлове да се генерират спрямо оригиналното чрез следния шаблон: "<original filename>.part<n>", където n е поредния номер на файла. Имената на новите файлове да се изпишат на екрана.

Примерен входен файл:	Примерни изходни файлове, limit = 100MB:
myfile.avi	myfile.avi.part00
	myfile.avi.part01
	myfile.avi.part02

20. Реализирайте програма, в която потребителя въвежда произволен низ, който съдържа букви и числа. Програмата трябва да анализира низа, да извлече числата от него и да изведе тяхната сума на екрана.

Примерен входен:	Примерен изход:
asd-12sdf45-56asdf100	77

21. Да се кодира даден низ използвайки Base64 алгоритъм за кодиране. За 62 и 63 символ да се използват символите '+' и '-', а за padding символа '='. За повече информация: http://en.wikipedia.org/wiki/Base64

22. Да се декодира даден низ, който е бил кодиран използвайки Base64 алгоритъм за кодиране. За 62 и 63 символ да се използват символите '+' и '-', а за padding символа '='. За повече информация: http://en.wikipedia.org/wiki/Base64

23. Потребител въвежда два низа: текст и ключ. Текста бива криптиран чрез ключа, като към всяка буква от текста, се прибавя стойността на съответната буква от ключа. Криптирания низ да бъде изведен на екрана, след което да бъде декриптиран обратно в оригиналния низ.

Примерен вход:	Примерен изход:
Text: ABCD	Encrypted text: ▒ г ╢ ╥
Key: pass	Decrypted text: ABCD

24. Потребителя въвежда масив с N на брой числа. Да се изведат последователностите от числа от тип: нарастващи, намаляващи, равни.

Примерен вход:	Примерен изход:
4 3 1 0 -3 -3 -3 5 6 7 3 3	4 3 1 0 - decrease -3 -3 -3 – evenly 5 6 7 – increase 3 3 - evenly

25. Потребителя въвежда 5 изречения с максимална дължина 50 символа. Да се изведат на екрана думите подред от 5-те изречения редувайки се: първа дума от първо изречение, втора дума от второ изречение... шеста дума от първо изречение следващата и т.н.

Примерен вход:	Примерен изход:
Hello world A B C D Fizz Buzz problem is cool	Hello A Fizz World B Buzz C Problem …

26. Потребителя въвежда 2 масива от по 10 елемента. Програмата проверява за всеки елемент от първия масив, дали има по-малък от него във втория. Ако има, разменя въпросните елементи на първия и на втория. След това записва втория масив в файл.

27. Потребител въвежда 10 числа в масив от 20 елемента. След това въвежда числа, като с всяко въведено число, указва индекса, където да бъде вмъкнато това число (избутва останалите). Въвеждането спира при въвеждане на 0 или запълване на масива.

Примерен вход:	Примерен изход:
5 1 3 6 2 6 // Първоначални данни 77 0 // Число, позиция 33 2 88 5 0	77 5 33 1 3 88 6 2 6