

# 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 в предвид 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 възпроизведат 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;

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 преименува файла на правилния extension и го прави.

Примерен вход:

photo.bmp

Примерен изход:

The correct filename extension should be png.

Should I correct the file name?

*Пояснение: всеки тип файл обикновено 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](http://en.wikipedia.org/wiki/Magic_number_%28programming%29#Magic_numbers_in_files)



14. Да се превърне \*.sub файл с MicroDVD формат в \*.srt файлов формат, т.е. от изменили 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:	Примерен изходен .srt файл:
{30}{50} Big ass title	1
{80}{120} Some actor name	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 борава 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 борава 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.

17. Нех 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	qŸ•yŸ™}iU oOъЪ

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 numrad блока на клавиатурата съответстват на възможните полета в играта. При всеки въведен ход, екрана се прерисува наново с попълнените полета.

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:

Text: ABCD

Key: pass

Примерен изход:

Encrypted text:  r || П

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:

4 3 1 0 -3 -3 -3 5 6 7 3 3

Примерен изход:

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:

Hello world

A B C D

Fizz Buzz problem is cool

Примерен изход:

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, заменя 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:

5 1 3 6 2 6 // Първоначални данни

77 0 // Число, позиция

33 2

88 5

0

Примерен изход:

77 5 33 1 3 88 6 2 6