## Задача 1:

Реализирайте командата **cat**. Програмата ви трябва да приема 0, 1 или повече аргументи, подадени от командния ред.

* 0 аргумента от командния ред: програмата чете редове от стандартния вход и изписва, това което е прочела на стандартния изход.

**Примерно извикване:** **./myCat**

* 1 аргумент от командния ред: програмата чете от файл, чието име е подадения аргумент. Програмата изпечатва съдържанието му на стандартния изход.

**Примерно извикване:** **./myCat file1**

**Примерен изход:** Изпечатва се съдържанието на file1.

* 2 или повече аргумента от командния ред. Програмата изпечатва съдържанието на файловете с имена, подадени като аргументи от командния ред последователно.

**Примерно извикване: ./myCat file1 file2 file3**

**Примерен изход:** Изпечатва се съдържанието първо на file1, после на file2, а накрая на file3.

## Задача 2:

Реализирайте програма, която приема произволен брой аргументи от командния ред, **сортира ги лексикографски** и ги записва в текстови файл. Името на файла, в който ще се записва, се въвежда от стандартния вход. Ако файлът не съществува - той се създава. Ако файлът съществува - изписва се съобщение за грешка.

**Примерно извикване:**

**./write\_args bbb ccc aaa abc**

**Примерен вход:**

**output1.txt**

**Съдържанието на output1.txt след работата на програмата:**

**aaa**

**abc**

**bbb**

**ccc**

## 

## Задача 3:

Даден е следния бинарен файл: <https://file.io/D5ZZg4G6AvQ0>.

В него е записана информация за 1000 студента. Информацията за един студент се пази със следната структура:

**struct Student {**

**char name[16];**

**double avg\_grade;**

**};**

Даденият файл е създаден с единствено извикване на **fwrite:**

**fwrite(s, sizeof(Student), 1000, f1);**

* **s** е масив с 1000 инициализирани елемента от тип **Student**.
* **f1** е указател към структура от тип **FILE.**

Реализирайте програма, която прочита съдържанието на файла и изпечатва информация за ученика с най-висок **avg\_grade.**

**Подсказки:**

* **Ползвайте fread за да прочетете информацията за всеки студент.**
* **ghex file\_name**

## 

## Задача 4:

Реализирайте програма, чрез която да се редактират бинарни файлове.

Програмата приема 1 аргумент от командния ред - името на файла, който ще се редактира.

Програмата може да изпълнява команди, прочетени от стандартния вход. Изпълнението на програмата протича по следния начин:

1. Отваря се файла, чието име е подадено като аргумент от командния ред. Ако файлът не съществува се създава. Ако файлът съществува - съдържанието му **НЕ се трие**.
2. Прочита се команда от стандартния вход. Възможните команди са:
   * **d - (display)** - Изпечатва се стойността на всеки байт от файла в шестнадесетична бройна система (подобно на данните в левия панел на ghex).
   * **t - (display text)** - Изпечатва се стойността на всеки байт от файла интерпретиран като ASCII символ (подобно на данните в десния панел на ghex).
   * **a <new\_byte> - (append)** - добавя нов байт със стойност **<new\_byte>** в края на файла. **<new\_byte>** е число в интервала **[0, ff]** в шестнадесетична бройна система.
   * **c <start\_index> <end\_index> - (cut)** - премахва байтовете, които са с индекси по-големи от **<start\_index>** и по-малки от **<end\_index>**. **<start\_index>** и **<end\_index>** са числа в десетична бройна система.
   * **e <index> <new\_byte> - (edit)** - замества стойността на байта с индекс **<index>** с **<new\_byte>**. **<index>** е число в десетична бройна система. **<new\_byte>** е число в шестнадесетична бройна система в интервала **[0, ff]**.
3. Програмата приключва при достигане на края на стандартния вход.

Направете 2 примерни входни файла за програмата:

* Първият входен файл съдържа команди, които да запишат текста **“Hello World!”** във файл.
* Вторият входен файл съдържа команди, които генерират файл, със съдържание, избрано от вас.

## 

## Примерно изпълнение:

**Стартираща команда:** **./my\_hexedit new\_file.bin**

| **Вход** | **Изход** |
| --- | --- |
| **a ff**  **a 10**  **a f5**  **d**  **a 12**  **d**  **c 1 2**  **d**  **a fe**  **a aa**  **a bb**  **a cc**  **d**  **c 3 4**  **d**  **e 2 00**  **d**  **e 3 aa**  **d**  **EOF** | **ff 10 f5**  **ff 10 f5 12**  **ff 12**  **ff 12 fe aa bb cc**  **ff 12 fe cc**  **ff 12 00 cc**  **ff 12 00 aa** |

## БОНУС 1: <https://docs.google.com/document/d/1MxChgDWMALOKvQGATrmG5ehpK5RsCydW3LataLVKY_M/edit?usp=sharing>