

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2022-2023 г.

Задачи за домашно № 3

1. Да се напише функция на C++, която приема като параметър символен низ S , с дължина не повече от 1024 символа и символ C . Функцията да криптира S , като за целта, C има ролята на ключ и всеки символ от S се криптира чрез операцията *побитово изключващо или* със символа C . Да се напише функция `main()`, която прочита от стандартния вход символ (ключ) и **един символен низ. масив-от-низове**. Програмата да отпечата на стандартния изход низовете криптирани с така написаната функция, според подадения ключ.

Примери:

Вход	Изход
! HelloWorld!	iDMMNvNSME

2. Даден е масив от цели числа, които са в интервала $[1, 9999]$. Нека наричаме числото, което е образувано от цифрите на последователно наредените му елементи *проекция на масива*. Така например проекцията на масива $[7, 2, 3, 51, 23]$ е числото 7235123. Да се напише функция F (изберете сами подходящо име на функцията) на езика C++, която приема като параметри масив от цели числа в интервала $[1, 9999]$, дължина на масива $N \in [1, 15]$ и пренарежда елементите му, така че проекцията на масива да има възможно най-голяма стойност. Реализирайте функция `main()`, която прочита от клавиатурата цяло число M , масив от M цели числа в интервала $[1, 9999]$ и извежда на екрана резултата от изпълнението на функцията F върху него. В случай на невалидни входни данни, програмата да извежда на екрана -1. **В случай че има няколко числа с еднакви цифри (например 8, 88, 888 и т.н.), те да бъдат подредени по големина във възходящ ред.**

Примери:

Вход	Изход
4	7 76 415 10

415 10 7 76	
6	9 8 52 19 12 1
8 9 12 19 52 1	

3. Поради увеличени брой полети напоследък, диспечерите на товарното летище в квартал Голямо Малово имат необходимост от помощ при определяне на необходимия брой площадки за престой на самолетите. Напишете програма, която по въведено разписание на полетите, отговаря на въпроса, колко е възможно най-малкият брой площадки за самолети, които трябва да бъдат осигурени на летището **за дадено денонощие**. Програмата трябва да приема като вход броят N на полетите за деня, последван от два реда, които представят съответно времената на излитане и кацане на самолетите от съответните полети, като на първия ред са посочени N часове на кацане, а на втория – N часове на излитане. Програмата да извежда на екрана на конзолата минималния брой площадки, който трябва да има на летището, както и началото и края на най-натоварения период за деня.

Пояснения:

- Приемаме, че което и да е време на пристигане, никога няма да съвпада с време на кацане (летището има само една писта).
- В даден момент всяка площадка може да бъде заета от не повече от един самолет.
- Всички времена на излитане и кацане (както за вход на програмата, така и за изход) са представени във вида "ННММ", където НН е час, а ММ – минути. **Първият възможен час (на кацане) за дадено денонощие е 00:01, а последният възможен (за излитане) - съответно 23:59.**
- Под *най-натоварен период* се разбира времеви интервал, в който на летището има едновременно най-много кацнали самолети. В случай че максималният брой кацнали самолети е еднакъв за повече от един времеви период, да се изведат всички периоди.

Примери:

Вход	Изход	Пояснение
6 0900 0940 0950 1100 1500 1800 0910 1200 1120 1130 1900 2000	3 1100-1120	В този период ще има три, едновременно кацнали самолета

4	2	В тези два периода ще има по два, едновременно кацнали самолета
0900 0905 1000 1110	0905-0910	
0910 0910 1120 1130	1110-1120	

4. Да се напише програма, която рисува в конзолата изображения съставени само от символите '#' и '.'. Обикновено в C++, символите се представят като числа, в интервала от 0 до 255, но тъй като в този случай може да има само два различни символа, за да се пести място, се използва по-различна схема, която ще обясним с конкретен пример.

Нека дадено изображение е представено с числата 60 39 136 и 0.

За да бъде разчетено то, ще преобразуваме числата в двоична бройна система:

00111100 00100111 10001000 00000000

След това ще разделим цифрите на групи от по 3 бита, от ляво на дясно:

001 | 111 | 000 | 010 | 011 | 110 | 001 | 000 | 000 | 000 | 00

В десетична бройна система, това са следните числа:

170236100000

За да се покаже изображението на екрана, се редува изписването на символите '.' и '#', като първото число (1) в редицата по-горе показва колко пъти се изписва символът '.', второто (7) – колко пъти се изписва символът '#', третото (0) - отново символът '.' и т.н. За отбелязване на край на изображението се използват две последователни нули в редицата, като останалите групи от три бита след тях се игнорират.

За да е напълно определено едно изображение, описано по този начин, се налага и отделно да се посочи неговата широчина в брой символи. Така например, при широчина 5, от редицата по-горе се получава следното изображение:

```
#####
#####
...##
#####.
```

Тъй като всяко три-битово число е между 0 и 7, за да се изпише 9 пъти поред символа '#', се използва поредицата "... 7 0 2 ...", т.е. 7 пъти '#', 0 пъти '.', 2 пъти '#'.

Да се напише програма на езика C++, която прочита от клавиатурата цяло число - широчина на изображение и поредица от числа между 0 и 255, които кодират

изображението по гореописания начин и след това “рисува” изображението на екрана. Въвеждането на числата да приключва след като в редицата от три-битови числа се срещнат две поредни нули. За да бъде валиден входът, данните трябва да запълват точно всички редове на изображението за дадената широчина.

Примери:

Вход	Изход	Пояснение
5 60 39 136 0	.#### ##### ...## ####.	Всеки ред има 5 символа. Крайната редица е 1 7 0 2 3 6 1 0 0: 1 път . и 7 пъти # 0 пъти . и 2 пъти # 3 пъти . и 6 пъти # 1 път . и 0 пъти # 0 пъти . (край след две поредни 0)
4 60 39 136 0	.### #### ##.. .### ###.	Същите символи, но разпределени по 4 символа на ред.
6 60 39 136 0	Invalid input	Получават се 20 символа, които не може да се разпределят на 6 реда (няма да има символи, с които да завърши последния ред).
14 179 131 193 188 54 84 79 40 146 81 34 212 85 69 199 28 113 199 28 113 193 48 194 154 104 128#####..... ...#####... ..#####.. .#.#####.####.##.....##. .##.#####.##. ###.#####.### ##.#####.#### ##### ##### ##### ##### #####.##### .##.....##..... .##.....##.....	Последните числа в редицата, съответстващи на последните няколко символа от изображението, са: 104 128 От тях последователно се получава: 01101000 10000000 011 010 001 000 000 0 3 2 1 0 0 Което е: 3 пъти . и 2 пъти # 1 път . и 0 пъти # 0 пъти . (край след две поредни 0)

Допълнителни инструкции за решаване и предаване на домашното:

1. При коректно решение, задача 1 носи 1 точка, задача 2 - 2 точки, а задачи 3 и 4 - по 3,5 точки.
2. Обърнете внимание, че в условието на всички задачи е посочено множеството от допустимите стойности на входните данни. Ако входът който се подаде е извън това множество, програмата трябва да изведе в конзолата следното съобщение: **"Incorrect input"** и да завърши своето изпълнение.
3. Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата.
4. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от преподавателите и при установено плагиатство ще бъдат анулирани.
5. **Всички решения на задачите от домашното трябва да бъдат предадени преди четвъртък, 22 декември, 23:59 ч.**
6. За решаване на задачите **не се допуска използване на STL функции.**
7. **Предадените от вас решения трябва да са написани на езика C/C++ и да могат да се компилират успешно на GCC.**
8. **Всяка задача от домашното трябва да бъде решена в точно един, отделен файл.** Името на файла трябва да бъде в следния формат:

fnXXXXX_d3_N.cpp, където:

- XXXXX е вашият факултетен номер
 - N е номерът на задачата.
9. Предаването на домашното се извършва в Github, според линка за предаване посочен в Moodle, като задачите се commit-ват в главната директория на repository-то. В Actions може да видите дали предадени задачи се компилират и дали примерните входове от условието се изпълняват успешно. След края на срока за предаване задачите ще бъдат тествани допълнително и с други тестове.
 10. Файловете с решенията, които предавате трябва да са оформени съгласно добрите практики за оформяне на кода, за които се говори по време на лекции и упражнения. Ще се отнемат точки за неинформативни имена на променливи, неизползване на подходящи константи и т.н.
 11. Файловете с решенията може да съдържат само стандартните ASCII символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
 12. Всички предадени програми трябва следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
 13. В началото на всеки един от файловете, които предавате, трябва има коментарен блок, който носи информация за съдържанието на файла. Този коментарен блок трябва да изглежда точно така, както е показано по-долу, като в него попълните

информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

```
/**
 *
 * Solution to homework assignment 3
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2022/2023
 *
 * @author <вашето име>
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 *
 */
```

Например един попълнен блок за студент с име Иван Иванов, ф.н. 12345, който предава задача 2, трябва да изглежда така:

```
/**
 *
 * Solution to homework assignment 3
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2022/2023
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 2
 *
 */
```

14. Предадени домашни, които не отговарят на условията от точки 5-12 от инструкциите описани по-горе ще бъдат оценени с 0 точки.