

## Бонус задачи за указатели, динамична памет и масиви.

**Задача 1.** Въведете число N, заделете динамична памет за масив от N цели числа. Въведете масива и след това намерете минималния елемент в него. Разменете местата на двете части - тази преди минималния елемент и тази, започваща от него до края. Ако има повече от един елемент с минимална стойност работете с най-левия.

**Пример:**

**Вход:**

9

2 6 1 3 0 1 3 2 9

**Изход:**

0 1 3 2 9 2 6 1 3

**Задача 2.** Фермерът Калоян ви е наел, защото има проблем с организирането на свинефермата си.

Той притежава огромно количество прасета - 100 000. Но те не са какви да е прасета. Те са свободно пасящи не-ГМО пшенични трици и био жълъди. Всяко прасе си има част от поляната, където може да се разхожда. Най-важното за прасетата е, че всяка вечер се нареждат в специално пригодени за тях кабинки където да си почиват и да спят. Там минава робот (Калоян има десетина такива), който им разказва приказка или изслушва тъгите от тежкия им прасешки живот. Обаче откакто фермата му достигнала такива размери, роботите му се оказали много бавни. Прасетата са недоволни, че не спят добре, напускат фермата му и отиват към други ферми.

Молим Ви, помогнете на Калоян!

Всяко от прасетата се характеризира с три елемента: **име**, **възраст** и **тегло** в килограми.

Вашата задача ще е в началото на програмата да попитате Калоян как иска да ги подреди: по възраст, по име или по килограми. След това ще изпрати един два робота за да потърсят определено прасе - по подадени килограми и възраст ще трябва да изведете името на прасето.

**Уточнение:**

Килограмите са число с плаваща точка в диапазона [10, 300]

Възрастта е цяло число (в месеци) между [1, 120]

Имената са символни низове с дължина до 15 символа

**Задача 3.** Въведете N. След него въведете N числа в интервала [0, 10000000]. След това ги разпределете в множество масиви, всеки от който съдържа елементите, чиито цифри се сумират до едно и също число (например 12001 и 31 ще са в една категория). Накрая ги изведете на екрана във формат: {сума}: {числа с цифри, сумиращи се до тази сума} (например: 4: 13 4 400 220)

**Забележка:** Опитайте се да използвате точна памет за масивите със сумите

**Пример:**

**Вход:**

9

187 4 15 18 22 38 42 36 54

**Изход:**

4: 4 22

6: 15 42

9: 18 36 54

11: 38

16: 187

**Задача 4.** Въведете числа N1, след което въведете N1 на брой цели числа - a1, a2 .. aN1, които ще бележим също като редица A. След това въведете число N2 и N2 на брой цели числа b1, b2 .. bN2 (тях пък ще наричаме редица B). Задачата ви е да конструирате нов масив, който съдържа всички елементи от редицата A и само тези от B, които не се делят на елемент от A. Накрая го изведете на екрана. Редът на елементите в резултатния масив не е важен.

*Забележка:* Опитайте се да използвате точна памет за масивите в програмата си.

**Пример:**

**Вход:**

5

15 4 17 25 39

9

2 18 40 13 150 51 39 300 16

**Изход:**

15 4 17 25 39 2 18 13