Семинарно 4

1. Всяка редица от равни числа в едномерен сортиран масив, се нарича площадка. Да се напише програма, която намира началото и дължината на най-дългата площадка в даден сортиран във възходящ ред едномерен масив(въвеждат се N и самият масив от потребителя).

Имаш сортиран масив (възходящ ред), т.е. имаш подредени числа в масив. Когато имаш еднакви числа, които се повтарят един до друг в масива (той е сортиран, така че няма нужда да го проверяваш) имаш площадка. Може да приемеш всяко число като различна площадка, дори и да не се повтаря. Търсиш дължината, т.е. броя повтарящи се еднакви числа в масива, и нейното начало, т.е. индекса на първия елемент от площадката. N е броят на елементите в масива.

Пример: 1, 2, 2, 3, 5, 5, 7, 7, 7, 8, 9

Площадки са:

* 1
* 2, 2
* 3
* 5, 5
* 7, 7, 7
* 8
* 9

Най-дългата площадка е 7,7,7. Дължината е 3, защото има 3 седмици, ,а индексът е 6.

1. Да се напише програма, която проверява дали между елементите на редицата a0, a1, ..., an са изпълнени релациите: a0 < a1 > a2 < a3…

(въвеждат се N и елементите на редицата)

Имаш едномерен масив от N елементи. Трябва да провериш дали членовете се редуват алтернативно спрямо големината им. Не знам дали алтернативно е така и на македонски, но идеята е да се редуват „<“ и „>“, т.е. нека първо е знакът „<“, след това ще е „>“, псоле „<“, след това „>“ и така се редуват. Задачата тук е да видиш дали масива изпълнява това условие: a0 < a1 > a2 < a3. Иначе казано като изключим проверката за първия елемент, защото няма предишен елемент, може да кажем, че трябва да проверим елементите от нечетен (odd) индекс е по-голям от предишния и да проверим дали четните (even) индекси са по-малки от предишния елемент на масива. Като приемам, че a0 е първия елемент, a1,3,5 са елементите с нечетни (odd) индекси и a2,4,6 са елементите с четните (even) индекси.

Пример:

Масив1: 3, 5, 4, 6, 5

Tук е изпълнено условието.

а0 = 3

а1 = 5

а2 = 4

а3 = 6

а4 = 5

a0 < a1 > a2 < a3

3 < 5 > 4 < 6 > 5

Масив2: 1, 3, 2, 4, 4

Тук **НЕ** е изпълнено условието, защото последните два елемента са равни.

Масив3: 1, 6, 4, 3

Тук **НЕ** е изпълнено условието, защото а2 = 4 е по-голямо от а3 = 3, а трябва да е по-малко.

1. Напишете програма, която получава масив от N елемента и обръща реда на елементите. Изкарайте ново полученият масив.

Имаш масив от елементи и трябва да принтираш масива, като първо трябва да го обърнеш.

Пример:

INPUT: 1, 4, 7, 3, 5, 6

OUTPUT: 6, 5, 3, 7, 4, 1

1. Напишете програма, която получава редица с цели N числа. След това получава цяло число К. Отместете всички елементи на редицата с K позиции, като използвате правилото за превъртане (когато елемент от последна позиция трябва да се измести то той трябва да отиде на първа позиция в редицата).

Пример:

INPUT: 1, 2, 3, 4, 5, 6, 7

K = 2

OUTPUT: 6, 7, 1, 2, 3, 4, 5

1. Дадена е редица с N цели числа. Да не намери K тия по големина елемент в редицата.

Тук трябва да направиш, така че да знаеш кои са най-големите по стойност елементи, т.е. да сортираш масива по-големина. Когато го сортираш да принтираш k-тия елемент от вече сортирания масив. Не се казва дали ако има два елемента от масива, които да са равни, дали трябва да се приеме като един или не, аз го направих да се приеме, но не е нужно.

Пример:

INPUT: 3, 5, 6, 2, 3, 4, 7

K = 3

OUTPUT: 5

Като сортираш масива става: 7, 6, 5, 4, 3, 3, 2. Третият елемент от сортирания масив е 5.

INPUT: 2, 7, 6, 6, 8, 8, 5, 4, 1

К = 2

OUTPUT: ако не приемаш, че има еднакви елементи (8 и 6 се повтарят), то изхода трябва да ти е 8, ако приемаш, че трябва да се игнорират повтарящите елементи, то изхода трябва да е 7.

Като сортираш масива става: 8, 8, 7, 6, 6, 5, 4, 2, 1

1. За дадена редица от числа да се намери и отпечата най-дългата (монотонно) нарастваща (или намаляваща) под редица

Тук ако разбираш условието не го чети, че мога да те объркам още повече. Ако искаш погледни примера само.

Същата като първа задача, само че тук трябва постоянно да проверяваш дали си в под редица в нарастващ (ascending) или в намаляващ (descending) ред. Не знам дали може да се застъпват елементите, според мен може ( ако последният елемент на под редицата, примерно от растящ (ascending) ред да стане първи от под редица с намаляващ (descending ) ред, ако елемента, след него е по-малък. Пример: 1, 2, 5, 4, 3, 4, 5. Тук може да приемаш, че под редиците са 1, 2, 5; 5, 4, 3 и 3, 4, 5. Тук 5 и 3 се „застъпват“ (елементът 5 го има в под редица 1 и 2, а елементът 3 го има в под редица 2 и 3. Трябва да запазваш максималния брой елементи на под редиците и техния първи елемент и като приключиш да принтираш най-дългата под редица.

Пример:

INPUT: 1, 4, 5, 7, 5, 3, 4, 5, 3

Под редици:

1, 4, 5, 7

7, 5, 3

3, 4, 5

5, 3

OUTPUT: 1, 4, 5, 7

Семинарно 5

1. Даден е двумерен масив NxM запълнен с цели числа. Проверете дали всеки ред съдържа числа, подредени в нарастващ ред, а всяка колона съдържа числа подредени в намаляващ ред.

Тук проверяваш първо редовете дали са в нарастващ (ascending) ред и после проверяваш колоните дали са в (descending) ред.

INPUT:

N = 4, M = 4

5, 6, 7, 8

4, 5, 6, 7

3, 4, 5, 6

2, 3, 4, 5

OUTPUT: Да редовете са в нарастващ ред, а колоните са в намаляващ ред.

INPUT:

N = 4, M = 4

1, 2, 3, 4

5, 6, 7, 8

9, 10, 11, 12

13, 14, 15, 16

OUTPUT: Редовете са в нарастващ ред, но колоните **НЕ** са в намаляващ ред.

2з. Даден е масив NxM запълнен с цели числа. Намерете числото, което има околни на него сбора, на които е най-голям. (околно е всяко число което се намира до, над, под или по диагонал на текущото на отстояние не повече от 1 поле).

Пример:

N = 4, M = 4

**3, 6, 5, 4**

**3, 4, 5, 2**

**7, 6, 4, 5**

**1, 2, 3, 8**

Търсим най-голямата сума на околните числа. Нека приемем, че сме стигнали до елемента **4,** тогава елементите в червено са неговите околни числа. Сбора е 3 + 6 + 5 + 3 + 5 + 7 + 6 + 4 = 39, така проверяваме за всички елементи и принтираме **елемента**, чийто околен сбор е най-голям.

3з. Дадена е матрица NxM запълнена с цели числа. Разменете редовете с най-голям и най-малък елемент.

Търсиш най-малкото число и най-голямото и разменяш редовете на които се намират. Ако са на един и същ ред, то масива не се променя.

Пример:

INPUT:

1, 2, 3, 4

5, 6, 7, 8

9, 10, 11, 12

13, 14, 15, 16

OUTPUT:

13, 14, 15, 16

5, 6, 7, 8

9, 10, 11, 12

1, 2, 3, 4

4з. В училището на Ели има 1 ≤ N ≤ 100,000 човека. Някои от тях се познават с други, като Ели знае, че съществуват 1 ≤ M ≤ 1,000,000 такива приятелства. Тя иска да напише програма, която бързо отговаря кои са приятелите на даден човек. Ако хората се подават с номера като цяло число, както и приятелите им изведете при поискан човек колко приятели има. (ако 1 е приятел с 3 то и 3 е приятел с 1).

Пишат се като вход между кои хора има приятелство.

INPUT:

N = 10, броят на хората

M = 5, брой двойки на приятели

1, 3

2, 5

2, 6

2, 7

6, 8

Приятели:

1: 3

2: 5, 6, 7

3: 1

4:

5: 2

6: 2, 8

7: 2

8: 6

9:

10:

Ако търсим кои са приятелите на 2, то резултатът трябва да е 5,6,7.

5з. Наум пише софтуер за малки деца, който има следните команди:

1: add(x) - добавяне на елемент Х към съществуващите до момента

2: delete(x) – изтрива всички срещания на елемента Х

3: smallest(x)  принтира Х тото най-малко число от наличните

Помогнете на Наум да напише своя софтуер, ако знаете, че Х е цяло число.

Тук трябва да използваш динамични масиви и да използваш realloc за промяна на размера на масива.

За delete(x) можеш да изместваш елементите наляво при всяко срещане на елемента X като почнеш от първия елемент

За smallest(x) трябва да копираш елементите от масива в друг и да го сортираш за да не промениш реда на елементите на първоначалния масив.