

**СУ „СВ. КЛИМЕНТ ОХРИДСКИ“, ФМИ**  
**Изпит по Увод в програмирането**

*спец. Компютърни науки 01.02.2019 г. Вариант Б*

**Задача 1.** (3 т.) Подредица на дадена редица  $a$  наричаме всяка непрекъсната последователност от елементи на редицата. Да се дефинира функция `unsigned progression(int a[], unsigned n)`, която намира дължината на най-дългата подредица от нарастващи числа в масива  $a$  с  $n$  елемента.

**Пример:** За масива  $\{1, 3, 1, 2, 3, 1\}$  дължината е 3, за масива  $\{4, 3, 2, 1\}$  дължината е 1.

**Задача 2.** (6 т.) Нека е дадена квадратната матрица  $A$  от цели числа  $\mathbb{N} \times \mathbb{N}$ ,  $2 \leq \mathbb{N} \leq 100$ , представляваща „лабиринт“. Елементите на матрицата със стойност по-голяма от числото 0 смятаме за „проходими“, а всички останали – за „непроходими“. „Низходящ път“ в лабиринта наричаме всяка последователност от проходими елементи на матрицата, в която всеки следващ елемент е или вдясно, или под предишния.

Да се дефинира функция `bool reachable(int A[][100], unsigned N, int sx, int sy, int target)`, която проверява дали в лабиринта, зададен чрез матрицата  $A$ , съществува низходящ път от елемента с координати  $(sx, sy)$  до някой елемент със стойност `target` такъв, че елементите на пътя да образуват ненамаляваща редица от числа.

Пример: На фигурата по-долу, такъв път съществува от елемента с координати  $(0,0)$  до целеви елемент със стойност `target=60`, но не и до целеви елемент със стойност `target=40`.

1	0	1	0
10	15	1	1
50	20	50	50
40	0	40	60

**Задача 3.** (6 т.) Да се дефинира подходяща структура `Planet`, описваща планетно тяло в система с една звезда със следните параметри (пренебрегваме влиянието на всички други тела, освен на звездата):

- координати на центъра на тежестта на тялото в евклидова отпавна система с начало в центъра на звездата на системата;
- диаметър;
- наличие на живот (да или не);
- описание (символен низ до 20 символа).

а) Да се създаде масив от 5 планети и да се въведат от стандартния вход стойности на параметрите им;

б) Да се дефинира функция `void printPlanets(Planet planets[], unsigned n)`, която по даден масив `planets` от  $n$  планетни тела извежда на

стандартния изход подходящо форматирана информация за планетите в реда на срещането им в масива;

в) Да се дефинира функция `Planet findPlanet(Planet planets[], unsigned n, char type)`, която в даден масив `planets` от `n` планетни тела намира и връща планетата, която:

- е най-близо до звездата на системата, ако `type=='s'`;
- е най-малката по диаметър и с наличие на живот, ако `type=='d'`, като ако на никоя от планетите в системата няма живот, да се върне планетата с най-малък диаметър;
- е с обем най-близък до средно аритметичното на обемите на всички планети, за всички други стойности на `type`.

Упътване: Допускаме, че планетите са сферични, обемът на сфера се намира по формулата  $V = \frac{4}{3}\pi r^3$ , където  $r$  е радиусът на сферата.