

Изпит по Увод в програмирането

спец. Компютърни науки, 1.02.2019 г.

Вариант А

Задача 1. (3 т.) Подниз на даден символен низ s наричаме всяка непрекъсната последователност от символи в s , включително празния низ и самия s .

Да се дефинира функция `unsigned longestSequence (char* s)`, която намира дължината на най-дългия подниз на s , който се състои от еднакви букви.

Пример: за низа "abbbcc", това е числото 3, а за низа "abc" — числото 1.

Бонус: (5 т.) За някой от циклите в решението (по Ваш избор) да се предложат подходящ инвариант и условие за частична коректност, като се скицира доказателство за верността им.

Задача 2. (6 т.) Нека е дадена квадратна матрица A от цели числа $N \times N$ ($2 \leq N \leq 100$), представяща "лабиринт". Елементите на матрицата със стойност по-малка от числото 10 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.

1	10	0	0
4	10	0	0
2	0	3	3
4	10	0	5

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 всички останали елементи от пътя да имат стойности по-малки от $target$.

Пример: На фигурата вляво такъв път съществува от елемента с координати $(0, 0)$ до целеви елемент със стойност $target = 5$, но не и до целеви елемент със стойност $target = 3$.

Задача 3. (6 т.) Да се дефинира подходяща структура `Body`, описваща физическо тяло със следните параметри (пренебрегваме влиянието на всички тела, освен на Земята):

- координати на центъра на тежестта на тялото в евклидова отправна система с начало в центъра на Земята;
- маса и скорост спрямо същата отправна система;
- описание (символен низ до 20 символа).

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

б) Да се дефинира функция `void printBodies (Body bodies[], unsigned n)`, която по даден масив `bodies` от n тела извежда на стандартния изход подходящо форматирана информацията за телата в реда на срещането им в масива.

в) Да се дефинира функция `Body findBody(Body bodies[], unsigned n, char type)`, която в даден масив `bodies` от n тела намира и връща тялото с:

- най-висока потенциална енергия, ако `type == 'p'`;
- най-висока кинетична енергия, ако `type == 'k'`;
- обща енергия (кинетична + потенциална), която е най-близка до средното аритметично на общите енергии на всички тела, при всички други стойности на `type`.

Ако има повече от едно тяло, отговарящо на условията, да се върне тялото с най-малък индекс в масива.

Упътване: Потенциалната енергия $E_p = mgh$, а кинетичната енергия $E_k = \frac{mv^2}{2}$, където m е

масата на тялото, $g \approx 9.8 \text{ m/s}^2$ е земното ускорение, h е разстоянието на тялото до Земята, а v е неговата скорост.

Изпит по Увод в програмирането

спец. Компютърни науки, 1.02.2019 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.

Бонус: (5 т.) за някой от циклите в решението (по Ваш избор) да се предложат подходящ инвариант и условие за частична коректност, като се скицира доказателство за верността им.

Задача 2. (6 т.) Нека е дадена квадратна матрица **A** от цели числа $N \times N$ ($2 \leq N \leq 100$), представяща “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.

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

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 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**.

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 отправна 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 информацията 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-близко 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

$$V = \frac{4}{3}\pi r^3, \text{ където } r \text{ е радиусът на сферата.}$$