



Софийски университет „Св. Климент Охридски“  
Факултет по математика и информатика

## Писмен изпит

курс Увод в програмирането  
за специалност Компютърни науки  
зимен семестър 2021/2022 г.

Основните критерии при оценяването на контролните ще бъдат:

- успешно изпълнение на поставеното условие;
- добро стилизиране и форматиране на кода;
- следване на добри практики за писане на код;
- **коректната работата с паметта.**

\* За решенията на задачите е позволено използването само на `<iostream>` и `<cstring>`

**Задача 1. (1,0)** Дадена е редица от цветни клетки с **максимален капацитет** 1024. Редицата се задава чрез едномерен масив от символи и съдържа три цвята – зелен, жълт и червен, които се задават с единичен символ, съответно 'g', 'y' и 'r'. Даден е робот, който може да се движи по цветните клетки само в права линия отляво надясно и разпознава цветовете на клетките, по които минава. Роботът прави последователни ходове, като спазва следните правила:

- при стъпване на зелен цвят, роботът преминава на следващата клетка; ако това е краят на редицата, роботът изпада в състояние на грешка, ходът се счита за неуспешен и не се зачита;
- при стъпване на жълт цвят, роботът продължава да се движи, докато спре на следващата зелена клетка; ако до края на редицата не намери зелена клетка, роботът изпада в състояние на грешка, ходът се счита за неуспешен и не се зачита;
- при стъпване на червен цвят, роботът приключва успешно работа.

В зависимост от коя клетка започва работа, роботът може да направи различен брой ходове. Да се реализира програма, която въвежда от стандартния вход данните за редицата, осигурявайки, че входът е коректен, и извежда на стандартния изход какъв е максималният брой ходове, който може да направи роботът в зададената редица.

**Примери:**

Редица	Резултат																				
<table><tr><td>r</td><td>g</td><td>y</td><td>r</td><td>g</td><td>g</td><td>r</td><td>y</td><td>g</td><td>r</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr></table>	r	g	y	r	g	g	r	y	g	r	0	1	2	3	4	5	6	7	8	9	<p>Отговор: 4</p> <p>Обяснение: започвайки от клетка с индекс 1, роботът прави ходовете: 1 → 2 → 4 → 5 → 6</p>
r	g	y	r	g	g	r	y	g	r												
0	1	2	3	4	5	6	7	8	9												
<table><tr><td>g</td><td>g</td><td>r</td><td>r</td><td>g</td><td>y</td><td>r</td><td>g</td><td>y</td><td>r</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr></table>	g	g	r	r	g	y	r	g	y	r	0	1	2	3	4	5	6	7	8	9	<p>Отговор: 3</p> <p>Обяснение: започвайки от клетка с индекс 4, роботът прави ходовете 4 → 5 → 7 → 8 → последният ход не се зачита, тъй като води до грешка</p>
g	g	r	r	g	y	r	g	y	r												
0	1	2	3	4	5	6	7	8	9												

**Задача 2. (1,5)** Даден е низ, който се състои от последователност от думи в произволен ред измежду следните – zero, one, two, three, four, five, six, seven, eight, nine, plus, minus **без интервали помежду им**. Низът описва аритметичен израз, в който всяка цифра и операция (събиране или изваждане) е описана със съответната ѝ английска дума. Всяко число се записва в израза като последователно се изписват всички негови цифри. Казваме, че низът е **коректен**, ако започва и завършва с название на цифра и в него няма две последователни названия на операции. Да се напише функция `calculateExpr`, която по подаден **коректен** низ от описания вид пресмята записания в него израз. Функцията да запише получения резултат във входния низ, на мястото на израза, под формата на число в десетична бройна система.

Да се реализира програма, която въвежда низ, с дължина **въведена** от стандартния вход, извиква функцията `calculateExpr` и извежда получения резултат на стандартния изход.

***Бонус (0.5):** Да се осигури, че подаденият низ е валиден и **коректен** спрямо горното описание.*

**Примери:**

Вход	Изход
onetwominusonethree	<p>"-1"</p> <p>Обяснение: Подаденият низ задава израз 12–13, резултатът е -1, представен като низ от три символа и записан на мястото</p>

	на подадения израз.
one minus two three plus four five six	<p>"434"</p> <p>Обяснение: Подаденият низ задава израза <math>1-23+456</math>, резултатът е 434, записан на мястото на подадения израз.</p>

**Задача 3. (1,5)** Ясен Яворов има малък бизнес за покупка и продажба на дърва за огрев. С камиона си той обслужва малкото село Квадратово, като разнася дърва за жителите му. Селото се слави с изключително правилната си архитектура: къщите в него образуват квадрат, като от всяка къща има улица към съседите ѝ на север, юг, изток и запад (освен тези в края на селото, разбира се). Всяка седмица Ясен получава заявки от съселяните си, които надлежно записва в компютъра си като квадратна матрица **map** от цели числа с размерност  $N \times N$  ( $N < 128$ ), в която клетките означават къщите, а числата означават заявките на всяка къща. Положително число **K** означава, че стопаните на къщата искат да продадат **K** кубика дърва, а отрицателно число **-L** означава, че те искат да купят **L** кубика дърва. Ясен иска да планира обиколката си като започне от къщата си на позиция (**homeRow**, **homeColumn**), да стигне до къщата на баба си (която също му е клиент!) на позиция (**grannyRow**, **grannyColumn**) и да се върне обратно в къщата си. Той се чуди дали има начин да направи това като:

- по пътя изпълнява точно заявките на къщите през които минава – понеже Ясен не иска клиентите му да се обидят и да го оценят с една звезда, той не може да ги подмине без да изпълни заявката им и винаги иска да я изпълни изцяло (т.е. за цялото количество);
- на връщане той не иска да минава обратно през клиенти, чиито заявки вече е изпълнил, за да оптимизира пътя си;
- числото в къщата на Ясен означава с колко кубика дърва той товари камиона си първоначално;
- не иска да претоварва камиона си повече от максималния му капацитет, зададен с положителното цяло число **maxVolume**;
- не иска да се окаже в ситуация, в която минава през къща, която иска да купи дърва, а той не носи достатъчно.

Помогнете на Ясен като реализирате функция **foundRoute**, която приема описаните по-горе параметри и връща като резултат булева стойност, указваща дали в селото има път, който удовлетворява изискванията на Ясен.

Упътване: използвайте рекурсия за търсене с връщане назад (backtracking) като записвате товара на камиона на всяка стъпка.

Бонус (0,5): Функцията да извежда и намерения път.

Пример	Резултат																									
<table><tr><td>1</td><td>5</td><td>6</td><td>4</td><td>-2</td></tr><tr><td>-4</td><td>10</td><td>4</td><td>-3</td><td>20</td></tr><tr><td>0</td><td>-4</td><td>2</td><td>-12</td><td>5</td></tr><tr><td>5</td><td>13</td><td>1</td><td>1</td><td>-7</td></tr><tr><td>2</td><td>7</td><td>5</td><td>0</td><td>15</td></tr></table> <p>Къщата на Ясен: (1, 1) Бабата на Ясен: (4, 3) Капацитет на камиона: 15</p>	1	5	6	4	-2	-4	10	4	-3	20	0	-4	2	-12	5	5	13	1	1	-7	2	7	5	0	15	true <p>Обяснение: Ясен може да тръгне по жълтия път и да се върне по червения път, като товарът му през цялото време е в допустимия капацитет:</p> $0 \leq 10 \leq 15$ $0 \leq 10 - 4 \leq 15$ $0 \leq 10 - 4 + 2 \leq 15$ $0 \leq 10 - 4 + 2 + 1 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 + 0 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 + 0 + 1 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 + 0 + 1 - 12 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 + 0 + 1 - 12 - 3 \leq 15$ $0 \leq 10 - 4 + 2 + 1 + 5 + 0 + 1 - 12 - 3 + 4 \leq 15$ <p>(в случая числото 0 в кафявата клетка означава, че бабата на Ясен няма заявка, а иска просто да го види)</p> <p>Бонус:</p> <p>(1,1) (2,1) (2,2) (3,2) (4,2) (4,3) (3,3) (2,3) (1,3) (1,2) (1,1)</p>
1	5	6	4	-2																						
-4	10	4	-3	20																						
0	-4	2	-12	5																						
5	13	1	1	-7																						
2	7	5	0	15																						