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

Домашна работа 3

курс Увод в Програмирането
за специалности Информатика и Компютърни науки
зимен семестър 2020/21 г.

Важно: Решенията ви трябва да отговарят и на дадените по-долу изисквания:

1. Решете задачите на C++;
2. Решенията ви трябва да се изграждат успешно и да работят с поне един от компилаторите GCC 8.1 или MSVC 19 (Visual Studio 2019) или по-стари техни версии.
3. Ако не е посочено друго, в решението можете да използвате готови функционалности от езика, които са включени в заглавните файлове <iostream>, <cmath>, <cstdlib>, <cstddef>, <cstring>, <new> и <limits>.
4. Където не е посочено друго, трябва да осигурите коректни входни данни.
5. Типовете на използваните променливи трябва да са съобразени с условието и със семантиката на програмата, която реализирате.
6. На всяко място, където използвате масиви и буфери, размерът им трябва да е точен спрямо нуждите ви.
7. Забранява се използването на глобални променливи.

Задача 1

С наближаването на сесията времето, което можете да отделите за подготовка за изпитите, намалява. Трябва да подредите задачите си по най-добрия начин, така че да постигнете оптимален резултат.

Входните данни във вашата програма са:

- Брой на задачите, които имате да изпълните - естествено число **tasksCount**
- Характеристиките за всяка задача - **tasksCount** тройки естествени числа, които съответстват на:
  - уникален индекс за задачата;
  - време (в минути), необходимо за изпълнението ѝ;
  - точки, които тя носи.
- Брой на дните, оставащи до сесията
- Поредица от минути - указват времето на ден, което можете да използвате за учене

Да се намери дали времето до сесията ще стигне за изпълнението на всички задачи. Ако това не е възможно, да се извършват максимален брой задачи за оставащото време, спрямо техния приоритет. Най-приоритетни са тези задачи, които носят най-много точки за най-кратко време. Ако една задача бъде започната, тя трябва да бъде завършена докрай и нейното изпълнение не може да бъде прекъснато.

Програмата извежда индексите на завършените задачи в низходящ ред според тяхната тежест и оставащото време за учене в часове и минути.

** Под тежест на задачата разбираме точките, които тя носи за даден период от време.*

Вход:
```
4
0 120 20
1 70 10
2 300 100
3 50 5
2
300
100
```
Изход:
```
Tasks: 2, 1
Time remaining: 0:30
```

Вход:
```
3
0 51 1000
1 50 600
2 50 600
1
100
```
Изход:
```
Tasks: 1, 2
Time remaining: 0:0
```

Вход:
```
2
5215 150 100
1002 80 40
6
10
20
40
30
70
60
```
Изход:
```
Tasks: 5215, 1002
Time remaining: 0:0
```

Вход:
```
5
0 20 40
1 60 30
2 80 80
3 450 500
4 200 40
6
120
10
500
300
150
80
```
Изход:
```
Tasks: 0, 3, 2, 1, 4
Time remaining: 5:50
```

Задача 2

Бикове и крави е логическа игра за отгатване на цифри. Играе се от двама противника, като всеки се стреми да отгатне тайното число, намислено от другия.

Играта протича по следния начин. На лист хартия всеки участник написва своето тайно число. Тайните числа са четирицифрени, като цифрите не трябва да се повтарят. След това, последователно един след друг, играчите задават въпрос с предположение за числото на противника. Противникът отговаря, като посочва броя на съвпаденията – ако дадена цифра от предположението се съдържа в тайното число и се намира на точното място, тя е „бик“, ако е на различно място, е „крава“.

Пример:
Тайно число: 4271
Предположение: 1234
Отговор: „1 бик и 2 крави“. (Бикът е „2“, а кравите са „4“ и „1“.)

Първият играч, който открие тайното число на противника, е победител.

Нека разгледаме следната модификация на играта:

- тайното число се състои от **4 различни** цифри от **1 до 9** (т.е. не използваме цифрата 0);
- само „първият“ играч намисля число, а „вторият“ се опитва да го познае;
- ако „вторият“ играч познае тайното число **с не повече от 7 хода**, то той печели играта. В противен случай „първият“ играч печели играта.

Асистентите имат своя програма, която „намисля“ тайно число и отговоря колко съвпадения имате.

Напишете функция `play`, която ще се компилира заедно с програмата на асистентите и ще играе срещу нея. Програмата на асистентите „намисля“ тайно число, а Вашата функция трябва да го отгатне. Ако спечели, то получава точките за съответния тест, а ако загуби, получава 0 точки за съответния тест.

Детайли по реализацията

Функцията `play` трябва да има следния прототип - `void play();`

Тя се вика веднъж от програмата на асистентите без аргументи. За комуникация с програмата на асистентите Ви е предоставена функцията:

```
const int* tryGuess(int);
```

Трябва да викате тази функция, за да играете поредния си ход, съобщавайки на противника вашето предположение за тайно число. Ако ходът Ви е валиден, функцията връща указател към масив от две числа: първото указва броя на „биковете“, а второто – броя на „кравите“ в предположението Ви. Ако функцията върне двойката (4, 0), значи сте победили. В този случай функцията `play` трябва да прекрати изпълнението си.

Ход е невалиден ако е изпълнено поне едно от следните условия:

- броят на цифрите на подаденото число е различен от 4;
- подаденото число е отрицателно;
- подаденото число съдържа повтарящи се цифри;
- подаденото число съдържа цифрата 0;
- функцията `tryGuess` бъде извикана отново, след като е върнала (4, 0).

При изиграване на невалиден ход или ако не познаете числото до 7 хода губите играта.

Предаване на решението

Вие трябва да предадете файл `BullsAndCows.cpp`, който съдържа функцията `play`. Той може да съдържа и други инструменти, необходими за работата на `play`, но не трябва да съдържа функция `main`. Той също така трябва да включва библиотеката `BullsAndCows.h` чрез указание към предпроцесора
`#include "BullsAndCows.h"` в началото си.
** За локално тестване можете да използвате допълнителен cpp файл.*

Примерна комуникация с програмата на асистентите

Тайното число на програмата на асистентите е 2471.

Номер на ход	**Ход**	**Върнат резултат**	**Обяснение**
1	**tryGuess(1234)**	**(0,3)**	Предположението съдържа три правилни цифри, но никоя не е на мястото си. Имате 0 бика и 3 крави.
2	**tryGuess(2813)**	**(1,1)**	Предположението съдържа две правилни цифри. Цифрата „2“ е на мястото си (бик), но цифрата „1“ не е (крава). Имате 1 бик и 1 крава.
3	**tryGuess(2741)**	**(2,2)**	Предположението съдържа четири правилни цифри. Две са на мястото си, две не са. Имате 2 бика и 2 крави.
4	**tryGuess(2471)**	**(4,0)**	Предположението е точно тайното число. Имате 4 бика и 0 крави. *В такъв случай функцията play трябва да прекрати изпълнението си.*

Задача 3

Дадена е следната дефиниция:
```
typedef unsigned char pixel[3];
```

Напишете функция
```
pixel** fillArea (
     const pixel *const *const image,
     size_t width, size_t height,
     size_t row, size_t column);
```

която получава като аргумент изображение (последователност от пиксели) през матрицата **image**. Изображението е съставено от **height** на брой реда, всеки с **width** пиксела. Пикселите са последователност от три байта - съответно интензитета на червения, зеления и синия цвят (RGB 24).
Целта на задачата е да запълни с черен цвят (пиксел със стойности {0, 0, 0}) областта, която съдържа пиксела с координати **row** и **column.** Една област се състои от съседни по хоризонтал или вертикал пиксели, средната интензивност на цвета на които се различава най-много с единица. Средната интензивност е средно-аритметично от интензивностите на трите цветови канала - (R+G+B) / 3.
В случай, че параметрите са коректни, функцията трябва да задели динамично нова матрица за пикселите, в която да съхрани модифицираното изображение. Ако параметрите не са коректни или възникне друг проблем, функцията трябва да върне като резултат **nullptr**.

Във вашата програма можете да напишете каквито искате помощни функции. Трябва да имате и кратка **main** функция, чрез която да покажете как се използва коректно вашата функция **fillArea.**

Пример:
За простота пикселите са представени като едно число - средния им интензитет.
Изображение с 5 реда и 6 колони.

```
20 20 20 15 18 10
22 21 20 19 18 14
20 20 26 16 22 42
21 25 25 20 19 41
19 20 23 43 44 44
```

Подават се координати 0, 0

Резултатно изображение:

```
 0  0  0 15  0 10
 0  0  0  0  0 14
 0  0 26 16 22 42
 0 25 25 20 19 41
19 20 23 43 44 44
```