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

Домашна работа 2

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>, <cstddef>, <cstring> 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.

Задача 1

Целта на тази задача е да реализирате конзолна версия на играта “Четири в редица”. Повече за нея можете да прочетете тук и тук.

Във вашата реализация размерите на дъската ще се въвеждат в началото на програмата. За да е валидна играта, както ширината, така и височината на игралното поле трябва да са поне 4, но за да можете да я изобразите добре на екрана, височината не е повече от 20, а ширината не повече от 40.

След това започва същинската игра, в която се редуват ходове на двамата играчи. Един след друг те трябва да въвеждат номер на колона в която да пуснат своите жетони. Програмата трябва да валидира коректността на хода, да изобрази на екрана текущото игрално поле и след това, ако играчът е спечелил, да обяви победата с подходящо съобщение. Тези стъпки се повтарят до края на играта (с победа на единия играч или приключване наравно).

От вас не се изисква да реализирате алгоритъм с който компютър може да играе срещу човек, но ако имате желание опитайте. Това може да донесе бонус точки.

Пример:
Където първите 2 числа са височината и ширината на дъската. А всяко следващо е стълбът, в който се пуска пулчето на играча, който е на ход.

```
5 4
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

1
| | | | |
| | | | |
| | | | |
| | | | |
|O| | | |

2
| | | | |
| | | | |
| | | | |
| | | | |
|O|X| | |

2
| | | | |
| | | | |
| | | | |
| |O| | |
|O|X| | |

3
| | | | |
| | | | |
| | | | |
| |O| | |
|O|X|X| |

4
| | | | |
| | | | |
| | | | |
| |O| | |
|O|X|X|O|
```

... След няколко хода:

```
4
| | | | |
| | | | |
|O|O|O|X|
|X|O|X|X|
|O|X|X|O|

4
| | | | |
| | | |O|
|O|O|O|X|
|X|O|X|X|
|O|X|X|O|

Player 1 wins!
```

Задача 2

Имплементирайте функция, която по зададени параметри да “изрисува” прогрес бар с ASCII символи, и да го съхрани в масив от символи. Параметрите, които трябва да приема функцията са: сегашен прогрес - число в интервала [0;1], ~~дължина на бар-а~~, затапващи символи (ляв и десен), булев флаг, указващ дали в бара да се показват процентите, запълващ символ и празен символ, целочислена бройка на деленията на целия бар. Преценете как да ги подредите, така че за някои от тях да можете да зададете смислени стойности по подразбиране. Преценете как да се получи резултатът.

Напишете програма, която прочита от потребителя цяло число - брой секунди и използвайки вашата функция “анимира” запълването на прогрес бара за тези секунди, като той започва от празен и стига до запълнен.

За постигане на по-добро усещане в тази задача е разрешено да използвате готовата библиотечна функция за приспиване на програмата ви:

```
std::this_thread::sleep_for(std::chrono::milliseconds(T));
```

за да накарате вашата програма да “чака” поне T милисекунди. Изберете подходящо време за чакане между извеждането на екрана и подходящо увеличение на прогреса, така че “анимацията” да бъде плавна.

Може да принтирате всяко ново състояние на нов ред, или да използвате някоя от функциите за изчистване на символите от конзолата, като например `system(“cls”)` за Windows или `system(“clear”)` за Linux/Unix. За тази цел също можете да включите подходящ заглавен файл.

Примерен изход в един момент, показващ прогрес бар със затапващи символи “[“ и “]”, запълващ символ “=”, празен символ интервал и показана стойност на процент на запълване.

```
[============    30%                    ]
```

Примерен изход със затапващи символи “<” и “>”, запълващ символ “-”, празен символ “.” и показване на процентите:

```
<-------------------------60%--....................>
```

Задача 3

Напишете програма, която дава възможности за работа с цели числа с голяма дължина - “дълги числа”. За представяне на числата трябва да използвате масиви с елементи от тип `char`, където всеки символ представя една десетична цифра. По желание можете да представяте числата и в друга бройна система, например 256-тична, в масиви от unsigned char, където всеки елемент представя една 256-тична цифра.

Вашата задача е да предоставите функции, чрез които може да се:

- прочете дълго число от клавиатурата. За определеност ще считаме, че всички числа, които се въвеждат, са коректни и нямат повече от 8192 цифри;
- изведе дълго число на екрана;
- сравняват две числа (за равенство и наредба);
- събират две числа;
- изваждат две числа;
- умножават две числа;
- **(бонус)** делят две числа (с частно и остатък).

Демонстрирайте вашата реализация чрез кратка програма, която прочита две такива числа и извежда на екрана резултатите от прилагането на всички изброени по-горе операции между тях - Дали са равни, кое е по-малко, тяхната сума, разлика и произведение. А ако сте написали деленето - съответното частно и остатък.

Пример:

Вход:

```
12345678901234567890
4354678097643135
```

Изход:

```
12345678901234567890 != 4354678097643135
12345678901234567890 > 4354678097643135
12345678901234567890 + 4354678097643135 = 12350033579332211025
12345678901234567890 - 4354678097643135 = 12341324223136924755
12345678901234567890 * 4354678097643135 = 53761457511741137249987999149935150
12345678901234567890 / 4354678097643135 = 2835
12345678901234567890 % 4354678097643135 = 166494416280165
```