Вариант А

Име: _____ ФН: _____ група: ___

Общи забележки:

- ➔ При реализацията на задачите си приемете, че входните данни са в коректен формат, но осигуряване на валидност на стойностите им е ваша грижа.
- ➔ Позволено е използване на преподаван на лекции материал, както и само на функции и средства от стандартната библиотека, декларирани в заглавните файлове `<iostream>` и `<cmath>`.
- ➔ Работете самостоятелно. Ако имате въпроси - попитайте квесторите в залата.

Задача 1:

Напишете програма, която прочита от клавиатурата 6 цели числа - **x1**, **y1**, **r1**, **x2**, **y2** и **r2**, които представляват координатите на центровете и радиусите на две окръжности. Програмата трябва да изведе в какво състояние се намират те (дали се пресичат, допират, влагат една в друга или нямат общи точки).

Задача 2:

Простите числа **p** и **q** наричаме числа близнаци, ако **q = p + 2**. Например **5** и **7** са числа близнаци. Да се напише програма, която прочита от стандартния вход цяло положително число **n** (n < 100000) и извежда на стандартния изход първите **n** двойки числа близнаци — всяка двойка на отделен ред, самите числа разделени с интервал. Припомняме, че 1 не е просто число.

Задача 3:

Една последователност от числа наричаме зиг-заг редица, ако всяко число е едновременно строго по-голямо или по-малко от непосредствените си съседи. Например последователността 1 3 2 3 1 7 4 е такава редица, а 1 5 4 3 6 не е.

- ➔ Да се реализира подходяща функция `isZigZag`, която проверява дали подаден като аргумент масив от цели числа е зиг-заг редица.
- ➔ Да се реареализира подходяща функция `makeZigZag`, която пренарежда подаденият ѝ като аргумент масив от цели числа, така че да стане редица зиг заг, ако това е възможно. Ако не е да съобщи с подходящо съобщение.
- ➔ Демонстрирайте използването на написаните функции в кратка програма.

Задача 4:

Напишете програма, която съдържа следните функции:

- `readArray`, която въвежда масив от **цели числа** от стандартния вход;
- `printArray`, която извежда масив от **цели числа** на стандартния изход;
- `isSorted`, която проверява дали подаден като аргумент масив от **цели числа** е сортиран (*възходящо или низходящо*);
- `countEvenEvenNumbers`, която преброява колко елемента в масива имат четен брой единични битове на четни позиции (*най-младшия е на позиция 0*);
- `clearArray`, която премахва елементите, преброени от предната функция, така че редът на останалите елементи да не се променя.

Демонстрирайте използването им в кратка програма.

Бонус задача:

Напишете програма, която прочита две цели числа **p** и **q** от стандартния вход и намира и извежда на екрана периодичното представяне на обикновената дроб **p/q**. Например, ако p = 14, а q = 9 трябва да се изведе 1.(5), ако p = 22, q = 7 трябва да се изведе 3.(142857), а ако p = 3, q = 4 да се изведе 0.75

Вариант Б

Име: _____ ФН: _____ група: ___

Общи забележки:

- ➔ При реализацията на задачите си приемете, че входните данни са в коректен формат, но осигуряване на валидност на стойностите им е ваша грижа.
- ➔ Позволено е използване на преподаван на лекции материал, както и само на функции и средства от стандартната библиотека, декларирани в заглавните файлове `<iostream>` и `<cmath>`.
- ➔ Работете самостоятелно. Ако имате въпроси - попитайте квесторите в залата.

Задача 1:

Напишете програма, която въвежда 6 цели числа - **x1**, **y1**, **r1**, **x2**, **y2** и **a**, които представляват координатите на център и радиус на кръг и на долния-ляв ъгъл на квадрат със страни успоредни на координатните оси и дължината на страната му. Програмата трябва да изведе в какво състояние се намират те (дали се пресичат, допират, влагат един в друг или нямат общи точки).

Задача 2:

Едно цяло число **X** наричаме задружно, ако може да се представи като сума на две различни прости числа. Ако едно задружно число е точен квадрат, тогава го наричаме двойно-задружно. Напишете програма, която при въведено число **n** *(n < 100'000'000)* намира и извежда на екрана всички двойно-задружни числа, по-малки от **n**. *Припомняме, че 1 <u>не е</u> просто число.*

Задача 3:

Ако за една редица sequence е изпълнено следното свойство:

`sequence[0] < sequence[1] > sequence[2] < sequence[3] >...` или

`sequence[0] > sequence[1] < sequence[2] > sequence[3] <...`

казваме, че това е редица-трион.

- ➔ Реализирайте подходяща функция isSawSequence, която по подадена редица проверява дали изпълнява горното условие.
- ➔ Реализирайте подходяща функция transformSequence, която пренарежда елементите на подадена като аргумент редица от цели числа, така че да стане редица-трион, ако е възможно.

Демонстрирайте използването на тези функции в кратка програма.

Задача 4:

Напишете програма, която съдържа следните функции:

- `readArray`, която въвежда масив от **числа с плаваща точка** от стандартния вход;
- `printArray`, която извежда масив от **числа с плаваща точка** на стандартния изход;
- `isSorted`, която проверява дали подаден като аргумент масив от **числа с плаваща точка** е сортиран възходящо;
- `numbersToClear`, която преброява колко елемента трябва да се премахнат, така че след това масивът да бъде сортиран в нарастващ ред. Първият елемент трябва да се запази;
- `clearArray`, която премахва елементите, преброени от предната функция, така че масивът да се получи сортиран в нарастващ ред.

Демонстрирайте използването им в кратка програма.

Бонус задача:

Напишете програма, която прочита две цели числа **p** и **q** от стандартния вход и намира и извежда на екрана периодичното представяне на обикновената дроб **p/q**. Например, ако p = 14, а q = 9 трябва да се изведе 1.(5), ако p = 22, q = 7 трябва да се изведе 3.(142857), а ако p = 3, q = 4 да се изведе 0.75