

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2022-2023 г.

Задачи за домашно № 2

Преди да предадете домашното си, прочетете внимателно инструкциите в края на този документ

1. Да се напише функция на езика C++ (с подходящо име), която реализира събиране и изваждане на особено големи числа. Функцията да приема следните параметри:

- Естествено число $N \in [1, 1024]$
- Първи операнд с дължина N цифри
- Естествено число $K \in [1, 1024]$
- Втори операнд с дължина K цифри
- Операция (събиране или изваждане)

Използвайте така написаната функция, за да реализирате функция `main()`, която прочита от клавиатурата две особено големи числа заедно с техните размери и отпечатва на екрана тяхната сума и след това - тяхната разлика. (За повече информация относно очаквания вход и изход вижте примерите). При некоректно въведени входни данни, **програмата да отпечатва на екрана "-1"**.

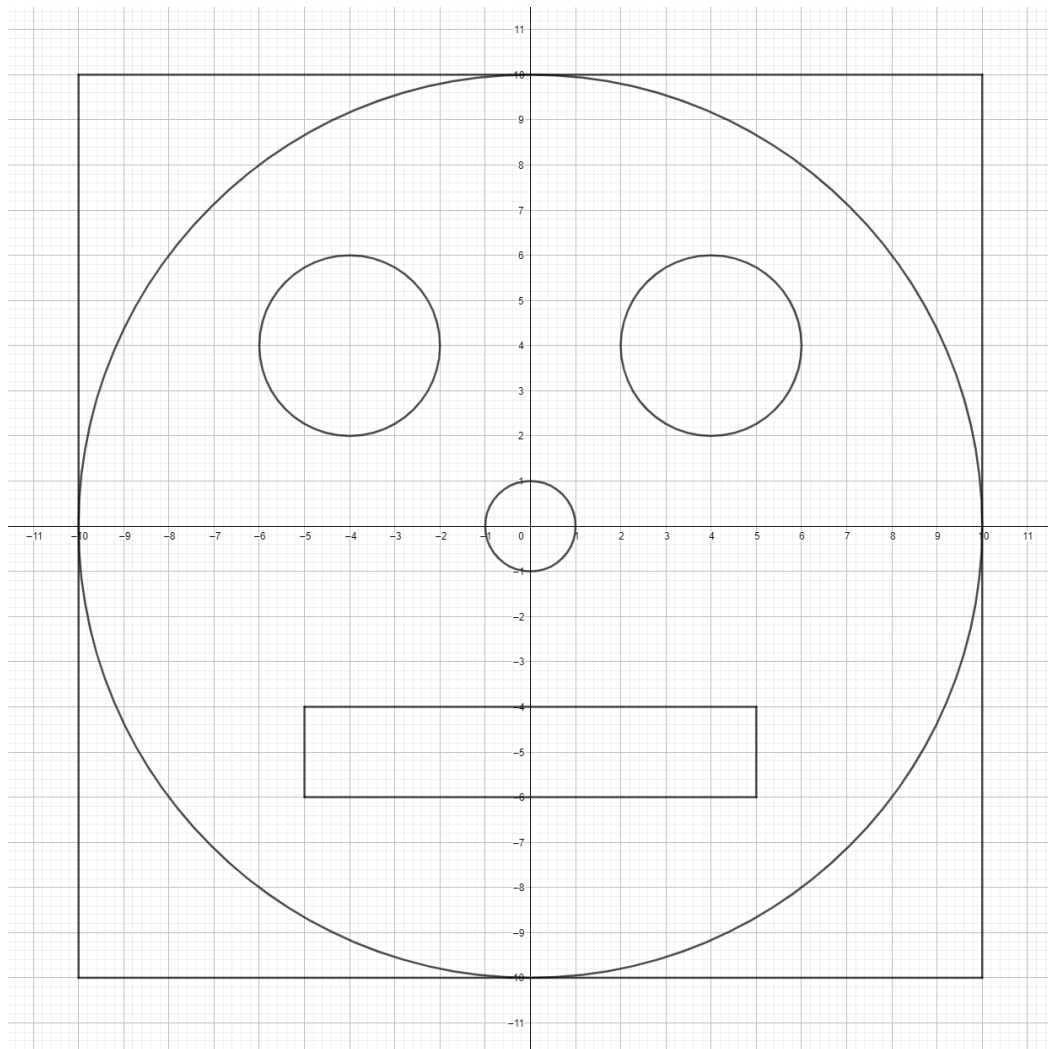
Примери:

Вход	Изход
14 12345678987654 13 1234567891234	13580246878888 11111111096420
3 999 3 999	1998 0

Пояснение: Вход в програмата може да бъде и отрицателно число, като в този случай в началото на входа има '-' (знак минус), а следващите символи са задължително цифри (най-много 1023). При изваждането винаги от първия се вади втория операнд.

2. Новият треньор на отбора по футбол в село Долно Нанагорнище решил да подобри ефективността на изстрелите от пряк свободен удар на футболистите си, като вместо във футболната врата, те трябвало да стрелят в специално подготвени за целта мишени. За повишаване на мотивацията, мишените имали формата на лицето на любимия футболен съдия на отбора, който никога не е отсъждал дузпа за тях в местната лига. Също така, мишените имат вградени сензори, които засичат къде е попаднала топката и подават данните като координати на точка в Декартова координатна система.

Помогнете на треньора да анализира напредъка на футболистите си като напишете програма, която при подадени от клавиатурата координати на точка в Декартова координатна система, връща съобщение за това, в коя зона е попаднала топката след поредния изстрел на играч на отбора.



Фиг. 1. Форма на мишената

Валидни са следните съобщения:

- Miss – ако няма попадение в рамките на мишената (квадрата)

- Background – ако попадението е в рамките на мишената, но извън целта (кръга)
- Face – ако попадението е в рамките на лицето, но извън очите, носа и устата
- Eye – ако попадението е в което и да е от очите
- Nose – ако попадението е в областта на носа
- Mouth – ако попадението е в областта на устата
- Border – ако попадението е върху някой от контурите

Пояснения:

Спецификациите на обектите на мишената са следните:

- Рамка на мишената – квадрат, определен от точките с координати: (-10.0, -10.0) и (10.0, 10.0)
- Лице – кръг с център с координати (0.0, 0.0) и радиус 10.0
- Очи:
 - Кръг с център с координати (-4.0, 4.0) и радиус 2.0
 - Кръг с център с координати (4.0, 4.0) и радиус 2.0
- Нос – кръг с център с координати (0.0, 0.0) и радиус 1.0
- Уста – правоъгълник определен от точките с координати (-5.0, -6.0) и (5.0, -4.0)

Абсцисната и ординатната оси са дадени само и единствено за ориентир и не са част от контурите.

Примери:

Вход	Изход
-2.0 4.0	Border
-5.0 0.0	Face
-9.0 9.0	Background
5.0 5.0	Eye
11.0 11.0	Miss

3. Нека съставим редица от цели числа, на която първия член е числото 1, а всеки следващ се описва от цифрите на предишния по следния начин:

- Първият член на редицата е числото 1
- Числото 1 се състои от 1 единица (1-1); вторият член на редицата е 11
- 11 се състои от 2 единици (2-1); третият член на редицата е 21
- 21 се състои от 1 двойка (1-2) и 1 единица (1-1); четвъртият член на редицата е 1211
- 1211 се състои от 1 единица (1-1), 1 двойка (1-2) и 2 единици (2-1); петият член на редицата е 111221
- Като се продължи по аналогичен начин получаваме следната редица:
1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, ...

Да се напише програма на езика C++, която приема като вход от клавиатурата цяло число $N \in [1, 20]$ и отпечатва на екрана в конзолата N -тия член на редицата. При некоректно въведени входни данни, програмата да отпечатва на екрана "-1".

Пояснение: Необходимо е да "изчисляват" членовете на редицата. Решения на задачата с оператор за условен преход, които изброяват всички възможни случаи от 1 до 20 ще се оценяват с нула точки.

4. Масив от цели положителни числа се нарича *магически*, ако съществува подмножество M от елементите на масива, такова че може да се образува полином, сумата на който е равна на единица, ако членовете му представляват произведение на елемент от M с произволно цяло число. Да се напише функция на езика C++ (с подходящо име), която приема като вход от клавиатурата дължина на масива $N \in [2, 1024]$ и елементите му - положителни цели числа, които принадлежат на интервала $[1, 10^9]$. Функцията да връща като резултат дали така въведеният масив е *магически*. При некоректно въведени входни данни, програмата да отпечатва на екрана "-1".

Примери:

Вход	Изход	Пояснение
4 5 12 7 23	1	$5 \cdot 3 + 7 \cdot (-2) = 1$
3 29 6 10	1	$29 \cdot 1 + 6 \cdot (-3) + 10 \cdot (-1) = 1$
3 102 207 24	0	

5. Нека разгледаме следните две операции, които са приложими върху множеството на целите естествени числа:

- Премахване на първата цифра на числото
- Премахване на последните две цифри на числото.

Да се напише функция на езика C++ (с подходящо име), която приема като параметър цяло естествено число $N \in [1, 10^9]$ и връща като резултат минималният брой от описаните по-горе операции, които са необходими за да се преобразува N до просто число. Ако това не е възможно, функцията да връща -1. При некоректно въведени входни данни, функцията да връща резултат -2.

Да се напише функция `main()`, която прочита от клавиатурата число $N \in [1, 10^9]$ и отпечатва на екрана резултата от изпълнението на така написаната функция върху него.

Примери:

Вход	Изход
202718	1
1031269	2
10271	0

6. Да се напише функция на езика C++ (с подходящо име), която приема като параметри две естествени числа $start$ и end , като $start, end \in [1, 2^{31}]$ и $start \leq end$. Функцията да връща като резултат броя на числата палиндроми и полупалиндроми в интервала $[start, end]$. При некоректно въведени входни данни, функцията да връща резултат -1.

Да се напише функция `main()`, в която се извиква написаната от вас функция за две числа въведени от клавиатурата и извежда на конзолата резултата от изпълнението ѝ.

Пояснение:

Палиндроми се наричат числа, които имат поне 3 цифри, и са еднакви прочетени отляво надясно и отдясно наляво (например 121 и 8998 са палиндроми). Полупалиндроми се наричат числа, които имат поне 3 цифри и те може да бъдат разместени така, че да се получи число палиндром (например 122 е полупалиндром, защото като се разместят цифрите му може да се получи 212, което е палиндром).

Примери:

Вход	Изход	Пояснение
120 135	4	121 и 131 са палиндроми, 122 и 133 са полупалиндроми.
1 100	0	Палиндромите и полупалиндромите са поне 3-цифрени числа. 100 не е полупалиндром, защото при разместване се получава 010, което не е валидно число.
121 121	1	

Допълнителни инструкции за решаване и предаване на домашното:

1. Задачи 1-4 се оценяват с 1,5 точки, а задачи 5 и 6 - с 2 точки.
2. Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата.
3. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от преподавателите и при установено плагиатство ще бъдат анулирани.
4. **Всички решения на задачите от домашното трябва да бъдат предадени преди събота, 3 декември 2022 г., 23:59 ч.**
5. За решаване на задачите **не се допуска използване на STL функции.**
6. **Предадените от вас решения трябва да са написани на езика C/C++ и да могат да се компилират успешно на GCC.**
7. **Всяка задача от домашното трябва да бъде решена в точно един, отделен файл.** Името на файла трябва да бъде в следния формат:

fnXXXXX_d2_N.cpp, където:

- XXXXX е вашият факултетен номер
 - N е номерът на задачата.
8. Предаването на домашното се извършва в Github, според линка за предаване посочен в Moodle, като задачите се commit-ват в главната директория на repository-то. В Actions може да видите дали предадени задачи се компилират и дали примерните входове от условието се изпълняват успешно. След края на срока за предаване задачите ще бъдат тествани допълнително и с други тестове.
 9. Файловете с решенията, които предавате трябва да са оформени съгласно добрите практики за оформяне на кода, за които се говори по време на лекции и упражнения. Ще се отнемат точки за неинформативни имена на променливи, неизползване на подходящи константи и т.н.
 10. Файловете с решенията може да съдържат само стандартните ASCII символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
 11. Всички предадени програми трябва следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
 12. В началото на всеки един от файловете, които предавате, трябва има коментарен блок, който носи информация за съдържанието на файла. Този коментарен блок трябва да изглежда точно така, както е показано по-долу, като в него попълните информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

```
/**
 *
 * Solution to homework assignment 2
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2022/2023
 *
 * @author <вашето име>
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 *
 */
```

Например един попълнен блок за студент с име Иван Иванов, ф.н. 12345, който предава задача 2, трябва да изглежда така:

```
/**
 *
 * Solution to homework assignment 2
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2022/2023
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 2
 *
 */
```

13. Предадени домашни, които не отговарят на условията от точки 5-12 от инструкциите описани по-горе ще бъдат оценени с 0 точки.