Общи забележки:

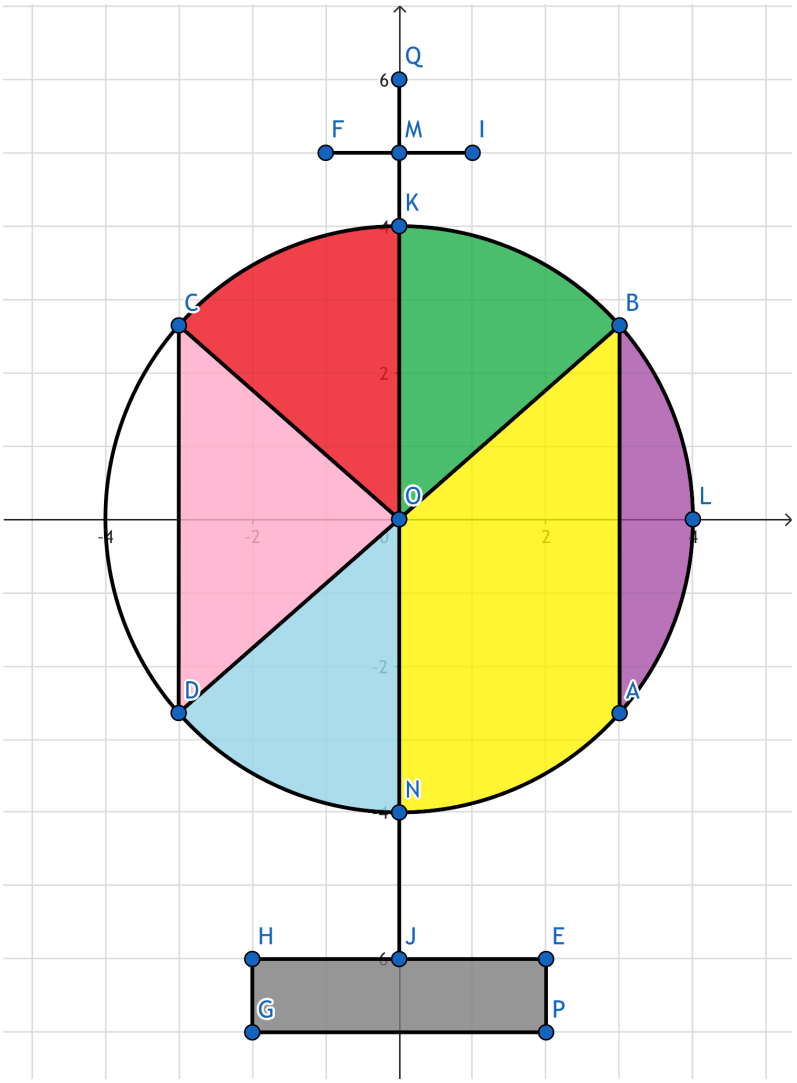
- При четене на входа, приемете, че потребителят въвежда данни в коректен формат и тип. Валидацията на стойностите им е ваша задача.
- При решението на тези задачи нямате право да използвате масиви и цикли.
- Разрешено е използването само на библиотеките **cmath** и **iostream**.

Задача 1

Дадена е следната фигура в Декартова координатна система. Да се напише програма, която прочита от клавиатурата координатите на точка (две числа с плаваща точка), определя в коя от оцветените области попада тя и извежда съответен текст за това на екрана:

- червената област (Red);
- зелената област (Green);
- розовата област (Pink);
- жълтата област (Yellow);
- синята област (Blue);
- лилавата област (Purple);
- сивата област (Grey).

Ако точката лежи на черните контури или на границата между две или повече области, счита се, че не е в нито една от тях и това също трябва да се съобщи на потребителя (On the edge). Ако е извън оцветените области и контура (т.е. в бялата област), също да се изведе подходящо съобщение (Outside).
Известни са координатите на точките:
H(-2, -6), E(2, -6), F(-1, 5), I(1, 5),
O(0, 0), L(4, 0) и абсцисите на
A(3, _), B(3, _), C(-3, _), D(-3,_).



Точността на работа на вашата програма трябва да е три знака след десетичната запетая.

Пример:

Вход:	Изход:
2 -2	Yellow

Вход:	Изход
0 -1	On the edge

Вход:	Изход:
4 4	Outside

Задача 2

Напишете програма, която прочита от стандартния вход цяло число без знак, което представлява часово време в двоичен формат, подобен на Binary clock. За улеснение ще считаме, че последните (най-младши) 6 бита на числото (битове от 0 до 5), взети като стойност, ще показват стойността на секундите, следващите 6 бита (от 6 до 11) - стойността на минутите, а битовете от 12 до 17 ще представят часовете. Бит 18 ще определя формата на часа. За разлика от минутите и секундите, часът се пресмята по следната схема: стойността, записана в битове 12, 13, 14 и 15, кодира втората цифра от часа (цифрата на единиците), а 16-ти и 17-ти бит - първата (десетиците). При стойност "1" на 18-ти бит ще считаме времето в 24 часов формат, а при "0" - в 12 часов. Всички останали по-старши битове трябва да са нулеви!

Вашата задача е да прочетете времето и да проверите дали е валидно. Ако е - изведете го във формат: HH:MM:SS.

Вход:	Изход:
14027	03:27:11
пояснение:	
14027 -> 0|00|0011|011011|001011-> 12 часов формат, час 0(00) 3(0011), минути 27 (011011), секунди 11 (001011) -> 03:27:11	

Вход:	Изход:
144705	Invalid data
пояснение:	
144705 -> 0|10|0011|010101|000001->12 часов формат, час 2(10) 3(0011), минути 21 (010101), секунди 1 (000001) -> 23 не е валидна стойност за 12 часов формат	

Вход:	Изход:
406849	23:21:01
пояснение:	
406849-> 1|10|0011|010101|000001-> 24 часов формат, час 2(10) 3(0011), минути 21 (010101), секунди 1 (000001) -> 23:21:01	

Задача 3

Вашата програма ще трябва да подпомогне работата на служителите в един магазин.
Всеки от продуктите в магазина има баркод, който кодира дата на производство и срок на годност (в дни). Колегите от Техническия Университет любезно са разработили баркод-скенера, който се справя с трудната част по излъчването на лазерни лъчи, приемането на отражението им от самия баркод, обработката на този сигнал, разчитането му и всички тежки хардуерни неща. В крайна сметка за вас (като по-вещи в програмирането) остава 'леката' задача да обработите едно цяло число (резултатът от сканирането) и да проверите дали съответният продукт е в срок на годност или не.
За удобство, приемете, че днес сме дата 01.11.2023, и заложете тази стойност като лесен за промяна, но известен по време на компилация параметър в програмата си.
Програмата, която трябва да напишете, трябва да прочита от клавиатурата едно цяло число. То ще описва датата на производство и срока на годност на даден продукт. Като отговор вие трябва да изведете един от следните три текста:

- ***Good*** - когато продуктът е в срок на годност (датата на производство с добавени съответния брой дни срок е до днешната дата, включително);
- ***Too old*** - когато продуктът е с изтекъл срок на годност (датата на производство с добавени съответния брой дни е след днешната дата);
- ***Invalid*** - когато подаденото число не описва коректна дата на производство.

Кодирането на датата е следното:
Най-младшите 5 бита описват деня от месеца, следващите 4 бита - месеца, следващите 9 бита описват годината като брой години след 1900. Следващите 13 бита описват срока на годност в брой дни.
Най-старшият бит играе роля на проверка по четност. Заедно с него всички единични битове в числото трябва да са четен брой. Така, ако данните за годността (дата на производство и срок на годност - младшите 31 бита на числото) са кодирани с използването на четен брой единици, то този бит трябва да има стойност 0. Ако тези данни са кодирани с нечетен брой единици - този бит трябва да е единичен и така да допълни броя единици до четно число.
При решението вземете предвид високосните години.
Помислете за различните варианти на некоректни данни и ги обработете съответно.

Вход:	Изход:
95745880	Good
пояснение:	
95745880 -> *0|0000101101101|001111011|1010|11000* *срок 365 дни, години 123, месец 10, дата 24* *Контролен бит 0* *Единици в данните 16 (числото е коректно)* *Дата на производство 24.10.2023* *Валиден до 23.10.2024 (2024 е високосна).* *Към дата 1.11.2023 е в срок*	

Вход:	Изход:
2243229016	Too old
пояснение:	
2243229016-> *1|0000101101101|001111010|1010|11000* *срок 365 дни, години 122, месец 10, дата 24* *Контролен бит 1* *Единици в данните 15 (числото е коректно)* *Дата на производство 24.10.2022* *Валиден до 24.10.2023.* *Към дата 1.11.2023 е с изтекъл срок*	

Вход:	Изход:
2243229528	Invalid
пояснение:	
2243229528-> *1|0000101101101|001111011|1010|11000* *срок 365 дни, години 123, месец 10, дата 24* ***Контролен бит 1*** ***Единици в данните 16 (числото не е коректно)***	

Вход:	Изход:
2243229656	Invalid
пояснение:	
2243229656-> *1|0000101101101|001111011|1110|11000* *срок 365 дни, години 123,* ***месец 14,*** *дата 24* *Контролен бит 1* *Единици в данните 17 (числото е коректно)* *Дата на производство 24****.14.****2023* *Датата не е коректна!*	