

Общи забележки:

- При четене на входа, приемерте, че потребителят въвежда данни в коректен формат и тип. Валидацията на стойностите им е ваша задача.
- Разрешено е използването само на библиотеките **cmath**, **ctime**, **cstdlib** и **iostream**.

Задача 1

Напишете функция, която приема като аргументи два масива от цели числа - А и В. Функцията трябва да проверява и връща като резултат дали масивът В може да се получи от масива А чрез премахване на един или повече от елементите му.

Демонстрирайте използването на функцията.

Вход	Изход
[1, 2, 4, 5, 2] [1, 5, 2]	true
[1, 2, 4, 2, 3] [2, 2, 4]	false

Задача 2

Предвид разочароващото представяне на националния отбор по футбол през последните години, селекционерът на триколюорите се явил пред БФС с "иновативна" идея. Треньорът твърдял, че е измислил начин да предотврати скатаването и ленивото движение на играчите по терена. Планът му бил прост и разумен — да се заложи на технологията. От БФС бързо приели, най-вече защото били чули, че така се прави "на запад" от доста време насам. Решено било в бутонките на футболистите да бъдат монтирани специални чипове, които да отчитат представянето на играча, който ги носи.

Тежката задача по реализацията на проекта възложили на студентите от ФзФ и ФМИ. За физиците не било никакъв проблем да направят интегрална схема на чипа, съдържаща GPS модул, памет и микроконтролер. След кратка дискусия, на която се обсъдило как да бъдат извличани данните от чиповете, единодушно се решило, че данните няма да бъдат предавани по мрежата (всички знаем колко несигурен е света на [IoT](#), а пък не искаме някой да краде информацията за това колко "всеотдайни" са нашите футболисти, нали?), а ще бъдат четени от паметта на чиповете след всяка тренировка / всеки мач.

На Вас се пада заключителната част на проекта — четенето на данните и преобразуването им в полезна информация. За целта напишете програма, която прочита от стандартния си вход: цяло положително число n ($1 \leq n < 2'000'000'000$) (броят на координатите записани в чипа), последвано от n реда, всеки от които с по две дробни числа с до пет значещи цифри (координатите, на които се е намирал играчът в дадения момент).

На стандартния си изход програмата извежда едно дробно число. При извеждането форматирайте изхода с три цифри след десетичната запетая. Числото представя общото разстояние, изминато от играча докато чипът е записвал.

Изминатото разстояние пресмятаме като сума от изминатите разстояния между всеки две последователни двойки координати. Тъй като играчите не се движат само по права линия, при пресмятане на разстояние между две точки трябва да отчетем и приблизителното им отклонение. За целта разглеждаме функция $f(x)$, която ни дава приблизителната дължина на разстоянието, изминато от играчите между две точки в равнината като приема за аргумент дължината на отсечката между тези две точки.

$f(x)$ се пресмята със следната формула:
$$f(x) = \frac{\frac{\pi x}{2} + x}{2}$$
, където за стойност

на константата π приемаме 3.14. Например, дължината на изминатия път между точките (0,3) и (4,0) е $f(5) = 6.425$.

Областта, в която ще следим за движението на футболистите ще бъде ограничена от игралния терен, зададен чрез правоъгълника **R** с върхове (-5,-2) и (5,2). За всички прочетени точки, които не попадат в тази област, ще спазваме следните правила:

1. Ако предходната точка **A** е вътре в терена, а текущата **B** е извън него, то при пресмятане на разстоянието вместо точка **B** използваме точката **B'**, която е пресечната точка на отсечката **AB** с правоъгълника **R** (т.е. не отчитаме разстоянието изминато извън терена).

Пример: Нека предходната точка **A** = (0,0), а текущата точка **B** = (0,3) (очевидно извън областта). Пресечната точка на отсечката **AB** и **R** е **B'** = (0,2). Изминатото разстояние между **A** и **B'** е $f(2) = 2.570$.

2. Ако и предишната, и текущата точка са извън терена, то разстоянието между тях не се прибавя към общото изминато разстояние. Тук приемаме, че дори отсечката между тях да пресича два пъти границите на игрището, то прекараното време на терена е незначително и не бива да калкулираме изминатото разстояние.

3. Ако предходната точка **A** е извън терена, а текущата **B** е в него, вместо точка **A** използваме точката **A'**, която е пресечната точка на отсечката **AB** с правоъгълника **R**.

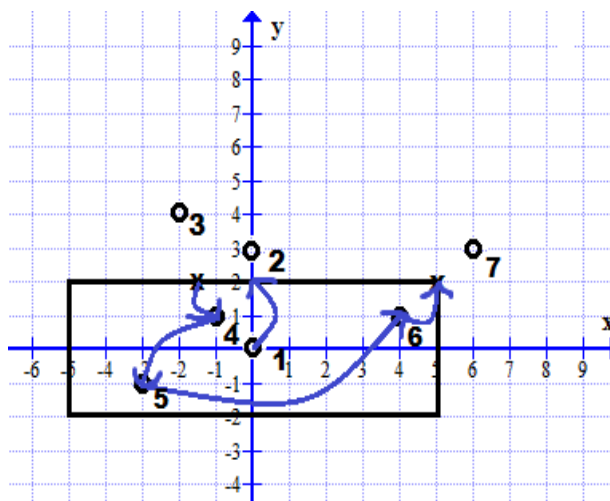
Пример: Нека **A** = (7,4) е извън терена, а **B** = (3,0) е вътре в терена. Пресечната точка на **AB** и **R** е **A'** = (5,2). Пресмятаме пътя между (3,0) и (5,2) по формулата $f(\sqrt{8}) = 3.635$.

4. Точки лежащи на контура на игрището приемаме за вътрешни. Използвайте за точност на сравненията само две цифри след десетичната точка.

Забележка: Забележете, че в дадените по-горе примери стойностите, върнати от f са закръглени до третата цифра след десетичната точка — тоест те са приблизителни. От вас се очаква да закръглите само крайния резултат, но НЕ и резултатите от междинните изчисления, понеже това би довело до допълнителна загуба на точност.

Пример:

Вход	Изход
7	18.731
0 0	
0 3	
-2 4	
-1 1	
-3 -1	
4 1	
6 3	



Приблизителна схема на движението на играча:

Задача 3

Всички добре познаваме играта [Бикове и Крави](#). В тази задача целта е да реализирате програма, която играе тази игра срещу потребител. При стартиране вашата програма трябва да си избере случайно число, което отговаря на правилата - естествено и четирицифрено, сред цифрите му няма повтарящи се. След това започва интерактивна игра срещу потребител. Той ще комуникира с играта през стандартните потоци за вход и изход. На всеки ход потребителят дава своето предположение за познаване на числото, избрано от вашата програма. Тя му дава информация за броя бикове и крави в това число. Следва неин ред - тя изказва своето предположение, след което потребителят отговаря с броя бикове и крави в намисленото от него число. Така ходовете се повтарят, докато някой от играчите познае числото на противника. Печели този, който познае пръв. Ако и двамата познаят в един ход - играта завършва наравно.

Позволете при стартиране да се указва дали да се играе само една игра (рунд) или няколко - например до достигане на 3 победи от единия играч.

Вашата програма трябва да следи за нарушения на правилата от потребителя - например ако неговото предположение не е коректно число да му съобщава това и той да губи хода си.

Опитайте да следите също за потенциално неверни отговори от страна на потребителя - например ако два пъти го попитате за едно и също предположение и той отговори различни неща. Във всяка ситуация, в която уличите в измама противника и обясните точно защо, печелите рунда служебно.

Забележка: За генериране на (псевдо)случайни стойности можете да използвате функцията [std::rand](#).