



Problema 1 – babaoarba

100 puncte

Tassadar joacă baba-oarba cu Zeratul și trebuie să îl găsească. Atât Tassadar, cât și Zeratul, se află la două puncte distincte din plan. Pentru a-l găsi pe Zeratul, Tassadar trebuie să se afle la o distanță de cel mult un metru de el.

Tassadar face succesiv pași de lungime de un metru, iar după fiecare pas, poate să pastreze direcția sau să efectueze un pas în altă direcție. Deoarece jocul ar fi prea greu pentru Tassadar, Zeratul îi va spune, după fiecare pas, dacă s-a apropiat sau nu.

Ajutați-l pe Tassadar să-l găsească pe Zeratul cât mai repede!

Interacțiune

Pentru a rezolva această problemă, va trebui ca în sursa voastră să existe următoarea funcție:

```
void play();
```

Această funcție va fi apelată o singură dată la începutul execuției programului. Din cadrul acesteia, puteți apela de câte ori doriți următoarea funcție, pusă la dispoziția voastră de către programul comisiei:

```
int makeStep(double direction);
```

Prin apelul acestei funcții, efectuați un pas de lungime un metru în direcția `direction` exprimată în radiani. Cu alte cuvinte, coordonata `x` la care vă aflați crește cu $\cos(\text{direction})$, iar coordonata `y` la care vă aflați crește cu $\sin(\text{direction})$. Această funcție va returna următoarele valori:

- -1, dacă în urma acestui pas vă aflați la o distanță de cel mult $1 + 10^{-9}$ de poziția la care se află Zeratul
- 1, dacă în urma acestui pas vă apropiați cu cel puțin 10^{-9} de poziția la care se află Zeratul
- 0, dacă nu vă aflați în niciunul dintre cele două cazuri anterioare

Odată ce funcția returnează -1, nu mai aveți dreptul să o apelați. În caz contrar, veți primi 0 puncte pe testul respectiv.

Restricții și precizări

- $1 < D \leq 100\ 000$, unde `D` este distanța la care se află Zeratul față de Tassadar în momentul inițial
- Pe toată durata jocului, Zeratul va sta nemișcat
- Punctajul acordat pentru un test va fi:
 - 100% din punctaj dacă apelați funcția `makeStep` de cel mult $\text{ceil}(D) + 50$ ori
 - 60% din punctaj dacă apelați funcția `makeStep` de cel mult $\text{ceil}(D) + 100$ ori
 - 20% din punctaj dacă apelați funcția `makeStep` de cel mult $\text{ceil}(1.5 * D) + 100$ ori
 - 0% din punctaj dacă apelați funcția `makeStep` de mai mult de $\text{ceil}(1.5 * D) + 100$ ori

Exemplu de interacțiune

Acțiuni grader	Acțiuni concurent	Explicație
<code>play()</code>	-	Inițial, Tassadar se află la coordonatele (0, 0), iar Zeratul la (2, 1.7). Tassadar
<code>return 1</code>	<code>makeStep(1.5707963)</code>	face doi pași spre Nord și se apropie de
<code>return 1</code>	<code>makeStep(1.5707963)</code>	Zeratul în ambele cazuri. Când efectuează al
<code>return 0</code>	<code>makeStep(1.5707963)</code>	treilea pas spre Nord, el se îndepărtează.
<code>return 1</code>	<code>makeStep(-1.5707963)</code>	Apoi, face un pas spre Sud și se apropie de
<code>return 1</code>	<code>makeStep(0.0)</code>	Zeratul. Urmează un pas spre Est și se
<code>return -1</code>	<code>makeStep(0.0)</code>	apropie de Zeratul, iar după al doilea pas
		spre Est, Tassadar ajunge la o distanță mai
		mică de un metru de Zeratul și îl găsește.

Timp maxim de execuție: 1 secundă/test.

Memorie totală disponibilă 16MB, din care 16MB pentru stivă

Dimensiunea maximă a sursei: 25KB