Bidding Game Challenge

Time limit: 10 seconds, Memory limit: 50 MB
Deadline for submissions: **2023-12-03**

Problem For this challenge, you will be playing a round-based auction game against all other participants simultaneously.

You start off with 10^5 coins. In each round, you bid a number of coins. Of course, you may only bid at most the amount of coins you have left. Everyone loses the number of coins they bid. The player with the highest bid wins the round and gets 1 point. In case of a tie, all participants with the highest bid split the point evenly.

You only know the bids of your opponents after you have made your bid.

After 10^3 rounds, all players are ranked by the number of points they earned (see Scoring for more details).

There will be 3 passes of the game as some solutions may use randomized strategies.

Interaction All interaction is done via standard input and output.

The first line of input contains a single integer n ($1 \leq n \leq 10^3$), the number of opponents.

Then for each round, you submit a bid by printing a single integer b ($0 \leq b \leq 10^5$) to standard output[1]. After making a bid, you will receive a line of input containing n space separated integers $b_1, \ldots, b_n$ ($-1 \leq b_i \leq 10^5$), the bid of your i-th opponent. A bid of -1 means that the opponent made an invalid bid or hit the time or memory limit. You are not required to read in the bids of your opponents after making your last bid.

If you make an invalid bid or hit the time or memory limit, you will be disqualified for the rest of the game. Your earned points will still remain valid.

See Sample Programs for an exemplary implementation.

Example interaction Below is an example interaction between four participants over three rounds each starting with 6 coins. On the left is the input given to the program, on the right is the output of the program.

[1] Remember to flush the output!

```
3
                                                                        0
2 2 4
                                                                        4
2 3 -1
                                                                        2
2 1 -1
```

Explanation. In the first round, the bids are 2, 2, 4 and 0. Therefore opponent 3 wins the round and gets 1 point. In the second round, the bids are 2, 3 and 4, so we win. Opponent 3 made an invalid bid and is disqualified from now on. In the third and final round, the bids are 2, 1 and 2, so we win together with opponent 1 and get half points each. Opponent 3 remains disqualified.

So we win with 1.5 points in total followed by opponent 3 with 1 point, opponent 2 with 0.5 points and opponent 1 with 0 points.

Appendix

Sample programs Here are two sample programs that play the game using a simple random and “copy the winner” strategy.

```python
# rand.py
import random
n = int(input())
coins = 10**5
rounds = 10**3

for round in range(rounds):
    my_bid = random.randint(0, coins)
    coins -= my_bid
    print(my_bid, flush=True)

    other_bids = [int(x) for x in input().split()]
```

```cpp
// copycat.cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    cin.tie(0)->sync_with_stdio(0);
    int n; cin >> n;
    int coins = 1e5, rounds = 1e3;
    vector<int> other_bids(n);
    for (int round = 0; round < rounds; round++) {
        int my_bid = clamp(
            *max_element(other_bids.begin(), other_bids.end()),
            0,
            coins
        );
        coins -= my_bid;
        cout << my_bid << endl;
        for (int &bid : other_bids) cin >> bid;
    }
}
```

Scoring Submissions are ranked by the following criteria, in order:

1. The number of points.
2. The number of rounds being not disqualified.

3. The number of coins left.

All numbers are calculated as the sum over all passes.

System Details All submissions are run on a single machine with the following specifications:

- Intel i7-11370H or better
- 16 GB RAM or more
- Debian GNU/Linux 12 (bookworm)

You may submit your solution as a single source file up to 64 kB in any of the following languages:

1. C++ 20 (g++ 10.2.1)

   compiled as `g++ -std=gnu++20 -x c++ -Wall -O2 -static -pipe -o $1 "$1.cpp" -lm`

2. Python 3.7.10 (PyPy 7.3.5)

   run as `pypy3 $1`

C++ solutions must compile within 30 seconds.

You may only use the standard library of your chosen language. You may not use

1. any other libraries or packages. Your submission must be self-contained.
2. any form of inter-process communication.
3. any kind of code obfuscation.
4. the internet.

Additional Rules

1. You may only submit one solution.
2. You may not exchange information about your solutions with other teams.