GS Lab Programming Contest #1

You are in Section 1: IPL Team Building, question 1 of 1. This is the last section.

IPL Team Building

For this contest, you have to write a program that is participating in a weird IPL team auction in which you have to build a team consisting of 11 players out of 40 available players. Based on past performance, your coach has assigned a specific number of points for all the 40 players. The points are an integer value from -9 to +9. The numeric value of the points indicates how good a player is. Players with negative points are actually considered to have a negative impact on the team (because they cause run outs, drop catches, and have behavioral problems) and are best avoided.

Each player is also categorized as a batsman, bowler, or a wicket-keeper.

Unfortunately, you can't acquire players one at a time - you have to acquire them in sets as follows. You have to write a function/method named `choose` which will be called by the system 40 times (once for each player - in random order) and your function needs to return one of these integer values:

- 1: this indicates "accumulate" *i.e.* add this player to the current set of players being considered. (The "current" set of players is empty at the beginning, and becomes empty after each "discard" that you do.)
- 2: this indicates "acquire" the entire set of players accumulated so far. All the players in the "current" set of players are added to your team.
- 3: this indicates "discard" the entire "current" set of players accumulated so far. These players will no longer be part of the auction, and the "current" set of players becomes empty.

Further rules regarding the auction are as follows:

- You can only "acquire" 4 sets. So you have to build your team with just 4 set acquires. In other words, the `yourMove` function/method can return `1` or `3` as many times as you wish, but you can return `2` only 4 times out of the 40 times it is called.

- It is legal to "acquire" a single player - but keep in mind the previous rule.
- Your 4th acquire must bring up your team size to exactly 11. Not having a team of size exactly 11 at the end of the auction results in a score of 0. Note: you may acquire 11 players in less than 4 sets - in this case, the game immediately stops and your team is scored.

Your team is scored as follows:

- All the points of your 11 players are added up, irrespective of the type of player he is. (Thus, negative players will actually reduce your score)
- If your team has more than 1 wicketkeeper, then 3 points are deducted from your team score for each *additional* wicket keeper. 10 points are deducted in case you have no wicketkeepers in the team.
- If your team has less than 4 bolwers, then 5 points are deducted for each bolwer you fall short. Thus 3 bolwers results in a 5 point deduction, 2 bowlers results in 10 point deduction, and 0 bowlers in a 20 point deduction.
- Similarly, if your team has more than 5 bowlers, 5 points are deducted for each additional bowler. Thus 6 bowlers result in a 5 point deduction, and an 11 bowler team will result in a 30 point deduction.

The aim of the game is to build a team with maximum points. The 40 available players consist of 18 batsmen with points going from -9 to +9 (one for each, excluding 0), 18 bowlers from -9 to +9 (excluding 0), and 4 wicketkeepers with +5 points each.

The system will run your program with various random (or carefully constructed not-so-random) sequences of the 40 players and the program with the best average score will be used to determine the winner.

Details of the code you have to write:

- In your program, you have to implement a class called `Game` which has one method called `yourMove`. (Separate instructions for C programs are given below)
- The `yourMove` method takes in a single string as its input argument
- The `yourMove` and returns a single integer as its result
- The evaluator will call `yourMove` exactly 40 times. Each time, it will send in a 3-character string as an argument. The first character of the input string is always `+` or `-`. The second character is a digit from `1` to `9`. The third character is either `T` (indicating batsman), `L` (indicating bowler), or `W` indicating a wicketkeeper. Some example values are: "+9T" (a batsman with value +9), or "-3L" (a bolwer with value -3), or "+5W" (a wicketkeeper with value +5).
- The `yourMove` method must return one of these values:
  - 1: indicating ACCUMULATE
  - 2: indicating ACQUIRE
  - 3: indicating DISCARD
- If at the end of 40 `yourMove` calls, your program has managed to acquire a team of exactly 11 players, using 4 or less ACQUIREs, then the team will be scored as described earlier. If the team size is not exactly 11 at the end of the 40th `yourMove` call, the score is 0.
- The evaluator will construct an instance of `Game` (by calling the constructor without any arguments) and will then call `yourMove` 40 times.
- For all programs, do not include the `main` program (or your language's corresponding equivalent) in the submission. Just submit your `Game` class. (To test your program, you will have to write your own test

cases, and main program, but please do not submit those.)

Instructions for specific languages: - For C, obviously there is no class. Instead you must implement the following: - A `struct Game` in which you can put whatever members you like - A `void constructGame(struct Game *g)` function that initializes an already allocated instance of a `struct Game` - A `int yourMove(char *)` as described above. - C programmers you may assume that the program will be compiled with the math library (`-lm`) but nothing else. - For Java, make sure that your class is a `public static void Game`, and make sure that there is no `package` declaration in your submission.

Languages supported (and the compiler we'll be using to compile/run the program):

- C - gcc 4.8.2
- C++ - g++ 4.8.2
- Java 1.6
- Java 1.7
- Java 1.8
- Ruby 1.9.3
- Python 2.7.6
- Perl 5.18
- PHP 5.5
- Bash 14.04
- JavaScript/NodeJS 0.10.25
- C# - C# 4.0 Experimental via mono mcs 3.2

The last date for submitting an entry is 17 October, midnight.

If you'd like another language added, or if you have questions about what's supported, please get in touch with **info@rsphinx.com**.

Answer

Warning: You will get only one attempt for this program, so make sure you submit a correct program. We recommend that you write the program in an IDE of your choice, compiled and test it, and then submit. You will not get a chance to correct a bad program for this question.

Submit Skip

- *Note: once you have skipped a question, you cannot return to it later.*

Powered by: