

Selecting a Majority Member

PROBLEM

Students in your school belong to two disjoint groups. It is known that one group, called the majority group, contains more students than the other group. We want to select a student who belongs to the majority group. Student's identifiers are integers from 1 to the number of students. The only operation that we can perform is to query whether two students belong to the same group.

You are to write a program that finds a student who belongs to the majority group, while it performs as few queries as possible.

LIBRARY

To perform queries, you are given a library query with three operations:

- **Size**, to be called once at the beginning, without arguments; it returns N , the number of all students. **Size** must be called before the first call to **Member**.
- **Member**, to be called with two student identifiers as arguments; it returns 1 if these two students belong to the same group, otherwise it returns 0.
- **Answer**, to be called once at the end, must be used to submit your solution. **Answer** has an integer argument, the label of a student belonging to the majority group.

The dialogue between your program and the library procedures is logged in the text file `select.out`. This log file also contains your answer and an indication whether your answer was correct.

Your answer is accepted only if for any two disjoint groups of all students A and B , for which all your queries give the same result, your answer identifies a member of the largest group of A and B . The library forces your program to issue all the queries that are necessary to identify a member of the majority group. That is, there are no pre-defined test cases but the library produces them "on the fly". If you give an answer before you are 100% certain that the answer is correct, you will get 0 points. So don't guess! ☺

Instruction for Pascal programmers: include the import statement

```
uses query;
```

in the source code.

Instructions for C/C++ programmers: use the instruction

```
#include "query.h"
```

in the source code, create a project file `select.gpr` in the task directory, add the files `select.c` (`select.cpp`) and `query.o` into this project, and then *compile* and/or *make* your program.

EXPERIMENTATION

You can experiment with the library by creating a text file `select.in`. The first and only line must contain the integer N , the number of all students. N must be an *odd* number!

EXAMPLE INPUT AND OUTPUT

select.in	select.out
7	Size=7 Member(1,2)=0 Member(3,4)=1 Member(5,6)=1 Member(4,6)=0 Your Answer: 7, Correct Majority Group: 2 5..7 Non-majority Group: 1 3 4 Number of Queries: 4 Full Possible Score: 3 Your Score: 3

For example, the answer 1 would not be accepted because, while all your queries give the same result for the groups {2, 5, 6, 7} and {1, 3, 4} 1 is not member of the majority group {2, 5, 6, 7}. (*a..b* denotes the set of integers from *a* to *b*.)

CONSTRAINTS

- For the number of students N , we have $5 \leq N < 30000$, and N is odd.
- Your program must not read or write any files.
- For student identifiers i , we have $1 \leq i \leq N$.
- FreePascal library file names: `query.ppw` and `query.ow`.
- Pascal function declarations:

```
function Size: integer;  
function Member(x, y: integer): integer;  
procedure Answer(x: integer);
```
- C/C++ library file names: `query.h`, and `query.o`
- C/C++ function declarations:

```
int Size(void);  
int Member(int x, int y);  
void Answer(int x);
```

GRADING

If your answer is correct and the number of queries that your program has performed is K then you obtain $\text{Max}(0, N-K)$ points, where N is the number of students.