Stage	1	2	3	4	Sum
Points	4	5	4	3	16
Result					

L2: Lab simulator

Your task is to write a program with a POSIX standard that simulates laboratory classes. There is the main process - a teacher and n child processes - students. During classes, students do the task divided into p parts ($1 \leq p \leq 10$). Each part of the task takes t times 100 ms to finish ($1 < t \leq 10$). So, in total, the task takes $p * t * 100$ ms.

The program takes as a positional argument parameters p, t and a list of numbers from $0 - 100$ (at least one). Each number determines each student's probability of having some issue during classes. It means that for every 100 ms of doing a task by a student, there is a given chance out of 100 that the student will need an additional 50 ms. So, students can have an issue t times per part of the task in the worst case.

When a student has finished some part of the task, raises a hand (sends `SIGUSR1` signal to the teacher) and waits for the teacher to check it. When the teacher notices a hand raised by a student, he checks the student's work and tells the student that he can continue (sends a `SIGUSR2` signal to a given student). When a student has finished all parts of the task, he exits (with a status equal to the number of issues he had during the task).

The teacher waits for all students to finish the task and then displays statistics about student issues formatted as below:

```
No. | Student ID | Issue count
  1 |      17487 | 3
  2 |      17488 | 6
 Total issues: 9
```

Use the following messages for appropriate actions:

```
Student[${no},${PID}] has started doing task!
Student[${no},${PID}] is starting doing part ${part} of ${total_parts}!
Student[${no},${PID}] has issue (${issue_count}) doing task!
Student[${no},${PID}] has finished part ${part} of ${total_parts}!
Teacher has accepted solution of student [${PID}].
```

A message starting with a "Student" must be printed by a student process and respectively "Teacher" by a teacher process.

HINT: To obtain signal sender `PID`, use a `siginfo_t` structure.

Example: `./sop-lsim 4 3 30 50` - runs a simulation with two students, which have 30% and 50% chances of having an issue each 100 ms of the task (which takes 4*3*100 ms in total without considering additional time per issue).

Stages:

1. 4 p. The teacher process starts students processes, children print on `stdout` their `PID` and probability value, then exit. Parent awaits children.
2. 5 p. The student process does the task according to the description and sends signals to the teacher (it does not wait for a response, just continues the work).
3. 4 p. The teacher process receives the signals from students and responds to them. The student waits for the task part confirmation.
4. 3 p. The teacher collects information about issues and prints them.

Example output:

```
$ ./sop-lsim 4 3 30 50
Student[0,18468] has started doing task!
Student[0,18468] is starting doing part 1 of 4!
Student[1,18469] has started doing task!
Student[1,18469] is starting doing part 1 of 4!
Student[1,18469] has issue (1) doing task!
Student[0,18468] has issue (1) doing task!
Student[1,18469] has issue (2) doing task!
Student[0,18468] has finished part 1 of 4!
Teacher has accepted solution of 18468 student.
Student[0,18468] is starting doing part 2 of 4!
Student[1,18469] has finished part 1 of 4!
Teacher has accepted solution of 18469 student.
Student[1,18469] is starting doing part 2 of 4!
Student[1,18469] has issue (3) doing task!
Student[0,18468] has issue (2) doing task!
Student[1,18469] has issue (4) doing task!
Student[1,18469] has issue (5) doing task!
Student[0,18468] has finished part 2 of 4!
Teacher has accepted solution of student [18468].
Student[0,18468] is starting doing part 3 of 4!
Student[1,18469] has finished part 2 of 4!
Teacher has accepted solution of student [18469].
Student[1,18469] is starting doing part 3 of 4!
Student[1,18469] has issue (6) doing task!
Student[0,18468] has issue (3) doing task!
Student[0,18468] has finished part 3 of 4!
Teacher has accepted solution of student [18468].
Student[0,18468] is starting doing part 4 of 4!
Student[1,18469] has issue (7) doing task!
Student[1,18469] has finished part 3 of 4!
Teacher has accepted solution of student [18469].
Student[1,18469] is starting doing part 4 of 4!
Student[0,18468] has finished part 4 of 4!
Teacher has accepted solution of student [18468].
Student[0,18468] has completed the task having 3 issues!
Student[1,18469] has issue (8) doing task!
Student[1,18469] has finished part 4 of 4!
Teacher has accepted solution of student [18469].
Student[1,18469] has completed the task having 8 issues!

No. | Student ID | Issue count
  1 |      18468 | 3
  2 |      18469 | 8
 Total issues: 11
```