3 questions   Total score: 30

3 Programming questions

1. Password Strength	+ 10.0
2. Raju's GATE coaching adventures	+ 10.0
3. Median of a list	+ 10.0

Question 1

Max. score: 10.00 ⊘

Password Strength

Let us say a password is *strong* if it contains a lowercase alphabet, an uppercase alphabet, a number, and a special character from `:`, `;`, `<`, `=`, `>`, `?`, or `@` (ASCII 58—64). For instance, the password "qwerty123" is not strong, while the string "@bC6ef" is indeed strong.

Write a program that reads a string of characters of length at most 64 characters from standard input and outputs "1" to standard output if the given string qualifies as a strong password, and outputs "0" to standard output otherwise.

Sample Input 1   Copy

```
qwerty123
```

Sample output 1   Copy

```
0
```

Explanation

Checks the four conditions for a strong password.

3 questions	Total score: 30

3 Programming questions

1. **Password Strength**	+ 10.0
2. **Raju's GATE coaching adventures**	+ 10.0
3. **Median of a list**	+ 10.0

ⓘ The following test case is one of the actual test cases of this question that may be used to evaluate your submission.

Sample Input 2   Copy

```
@bC6ef
```

Sample output 2   Copy

```
1
```

Note: Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Time Limit: 5.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Score is assigned if any testcase passes
Allowed Languages: C, C++, Java

Question 2   Max. score: 10.00

Raju's GATE coaching adventures

Raju is preparing for GATE to join IISc. He has joined Mogambo's GATE coaching institute that has given him several chapters of notes to study. The chapters are numbered using integers. There are dependencies between the chapters, expressed as "N -> M", which means that chapter number N depends on chapter number M. This dependency means that Raju must first study chapter number M before studying chapter number N.

Unfortunately, Mogambo's GATE coaching institute has not done a good job in preparing the chapters. There may be *circular* dependencies between the chapters, which would mean that Raju cannot study using these notes.

Raju wishes to efficiently determine whether there is a circular dependency in the chapters given to him that would prevent him from making use of the notes. Write a program to help Raju make this determination.

INPUT: The input to your program will be a set of dependencies between the chapters, with each dependency given on a single line N -> M, on standard input. The chapter numbers need not be in serial order. You should also not expect to see all chapters from 1 onwards, since there may be missing chapters in the set of notes. You should also not make any assumptions that if N -> M is listed as a dependency, then integer M will be numerically smaller than integer N.

TIME LIMIT: The correctness of your algorithm using 10 test cases. You will receive 1 point for every test case for which your algorithm produces the correct answer. Furthermore, your algorithm must be able to compute the answer to the problem **within 5 seconds** of wall-clock time on our server for each of these test cases. If your algorithm times-out on our test server for one of our test cases, then you will receive 0 points for that test case.

OUTPUT: Your program should output 1 if there is a circular dependency between the chapters, and 0 otherwise.

Sample Input 1   Copy

```
100 -> 6
99 -> 7
34 -> 55
6 -> 55
7 -> 74
55 -> 16
16 -> 100
```

Sample output 1   Copy

```
1
```

Explanation

Contains a cycle 100 -> 6 -> 55 -> 16 -> 100

3 Programming questions

1. Password Strength	+ 10.0
2. Raju's GATE coaching adventures	+ 10.0
3. Median of a list	+ 10.0

ⓘ The following test case is one of the actual test cases of this question that may be used to evaluate your submission.

Sample Input 2   Copy

```
100 -> 6
99 -> 7
34 -> 55
6 -> 55
7 -> 74
55 -> 16
```

Sample output 2   Copy

```
0
```

Note: Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Time Limit: 5.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Score is assigned if any testcase passes
Allowed Languages: C, C++, Java

Median of a list

You are given a large list of integers (you don't know size of the list *a priori*). The integers are given to you one at a time, each on a single line, on standard input. An integer may appear multiple times in this list. Write a program to find the **median value** of the given list of integers. The median value of the list is defined as follows:

- Arrange the data points in increasing order from smallest to largest.
- If the number of data points is odd, the median is the data point in the middle position of the list.
- If the number of data points is even, the median is the average of the data points in the two middle positions of the list.

Output the median value on standard output, with a single decimal precision (eg 5.5).

Sample Input 1   Copy

```
10
1
2
9
8
7
3
6
5
4
```

Sample output 1   Copy

```
5.5
```

1. **Password Strength**   + 10.0

2. **Raju's GATE coaching adventures**   + 10.0

3. **Median of a list**   + 10.0

Explanation

Outputs the median value of the list.

ⓘ The following test case is one of the actual test cases of this question that may be used to evaluate your submission.

Sample Input 2   Copy   **Sample output 2**   Copy

```
10
20
30
```

```
20.0
```

Note: Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Time Limit: 5.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Score is assigned if any testcase passes
Allowed Languages: C, C++, Java