ACM ICPC

UM Practice Contest 1

September 16

Problems

A A+B Problem
B Fibonacci Numbers
C Range Sums
D Drainage Ditches

Problem A
A+B Problem

Description

Calculate $a + b$

Input

Two integers a, b $(0 \leq a, b \leq 10^9)$

Output

Output $a + b$

Sample Input

1 2

Sample Output

3

Hint

Q: Where are the input and the output?

A: Your program shall always read input from stdin (Standard Input) and write output to stdout (Standard Output). For example, you can use 'scanf' in C or 'cin' in C++ to read from stdin, and use 'printf' in C or 'cout' in C++ to write to stdout.

You shall not output any extra data to standard output other than that required by the problem; otherwise you will get a "Wrong Answer".

User programs are not allowed to open and read from/write to files. You will get a "Runtime Error" or a "Wrong Answer" if you try to do so.

Here is a sample solution using C++/ G++:

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b;
    cout << a+b << endl;
    return 0;
}
```

It's important that the return type of main() must be int when you use G++/ GCC, or you may get compile error.

Here is a sample solution using C/ GCC:

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a + b);
    return 0;
}
```

Here is a sample solution using Java:

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String args[]) throws Exception {
        Scanner cin = new Scanner(System.in);
        int a = cin.nextInt(), b=cin.nextInt();
        System.out.println(a + b);
    }
}
```

Problem B

Fibonacci Numbers

Description

The Fibonacci numbers are the numbers in the following integer sequence, called the Fibonacci sequence, and characterized by the fact that every number in it is the sum of the two preceding ones:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Let f[n] be the n-th Fibonacci number. We know that f[0] = 0, f[1] = 0. Find f[n].

Input

An integer n.

For problem B1, $0 \leq n \leq 10^6$.

For problem B2, $0 \leq n \leq 10^{18}$.

Output

We know that f[n] is going to be very big, so instead of f[n], output f[n] mod 1000000007.

Sample Input 1

10

Sample Output 1

89

Sample Input 2

50

Sample Output 2

365010934

Note

In the real contest, each problem has only one size constrain. In this practice contest, many problems have two, one small (B1) and one large (B2), because it’s a good way to practice.

Problem C

Range Sums

Description

We want to maintain a list of n integers: $l[1], l[2], \ldots, l[n]$. At the beginning, the n integers are given. After that, there is going to be m operations, each operation would either be an update: ask for updating an integer, or query: ask for the sum of a range. For each query, output the sum of the range.

Input

First line contains two integers n, m, which represents the list size, and the number of operations.

Second line contains n integers: $l[1], \ldots, l[n]$, in which $0 \leq l[i] \leq 10^8$.

The following m lines contain m queries. Each line contains three integers: a, b, c. Each line starts with either 1 or 2 (a = 1 or 2).

If $a = 1$, then it is an update operation: increase $l[b]$ by c. That is, $l[b] = l[b] + c$. It is guaranteed that $1 \leq b \leq n$, and $0 \leq c \leq 10^8$.

If $a = 2$, then it is a query. Output $l[b] + l[b + 1] + \ldots + l[c]$. It is guaranteed that $1 \leq b \leq c \leq n$.

All inputs are valid, so that a could not be any other value.

For problem C1, $1 \leq n, m \leq 2000$.

For problem C2, $1 \leq n, m \leq 200000$.

Output

For each query, output the sum of the range.

Sample Input

```
5 5
1 2 3 4 5
2 1 5
1 2 3
1 3 4
1 4 5
2 1 5
```

Sample Output

```
15
27
```

Problem D

Drainage Ditches

Description

Every time it rains on Farmer John's fields, a pond forms over Bessie's favorite clover patch. This means that the clover is covered by water for awhile and takes quite a long time to regrow. Thus, Farmer John has built a set of drainage ditches so that Bessie's clover patch is never covered in water. Instead, the water is drained to a nearby stream. Being an ace engineer, Farmer John has also installed regulators at the beginning of each ditch, so he can control at what rate water flows into that ditch.

Farmer John knows not only how many gallons of water each ditch can transport per minute but also the exact layout of the ditches, which feed out of the pond and into each other and stream in a potentially complex network. Given all this information, determine the maximum rate at which water can be transported out of the pond and into the stream. For any given ditch, water flows in only one direction, but there might be a way that water can flow in a circle.

Input

The input includes several cases. For each case, the first line contains two space-separated integers, N (0 <= N <= 200) and M (2 <= M <= 200). N is the number of ditches that Farmer John has dug. M is the number of intersections points for those ditches. Intersection 1 is the pond. Intersection point M is the stream. Each of the following N lines contains three integers, Si, Ei, and Ci. Si and Ei (1 <= Si, Ei <= M) designate the intersections between which this ditch flows. Water will flow through this ditch from Si to Ei. Ci (0 <= Ci <= 10,000,000) is the maximum rate at which water will flow through the ditch.

Output

For each case, output a single integer, the maximum rate at which water may emptied from the pond.

Sample Input

```
5 4
1 2 40
1 4 20
2 4 20
2 3 30
3 4 10
```

Sample Output

```
50
```