Indian National Olympiad in Informatics, 2015

Time: 3 hours   *31 January, 2015*

Instructions

(a) You will have to return this question paper at the end of the examination with relevant parts filled out.

(b) There are two questions. You have to write working programs in C, C++, Java or Pascal to solve each of these questions.

- *Only your source code will be submitted for evaluation. Your program will be recompiled and run on the evaluation computer.*

- *Make sure your C/C++ programs compile with the GNU C compiler (**dev-cpp** or **Code::Blocks** under Windows). Programs written using Turbo C++ may not compile and run in the evaluation environment and may hence lose marks.*

  If you work in C/C++, make sure you do not write #include <conio.h> in your program or use any functions defined in conio.h. **If you do so, your program will not compile and you will get no marks.**

(c) At the end of each question, there is a space to indicate the location of the source code file for your solution. Please fill up this information without fail. Otherwise, your solution cannot be evaluated.

(d) All input for your programs will come from the keyboard. All output from your programs should be written to the screen.

(e) *On the computer where you are working, you will find a collection of sample inputs and outputs for each problem to test your programs. Please ask your centre supervisor if you cannot locate these sample inputs and outputs.*

   This set of sample inputs is not exhaustive. Additional test cases will be used for final evaluation. Even if your program correctly solves all these sample inputs, there is no guarantee that your program will pass all the cases in the final evaluation.

(f) Please fill out your contact details on the reverse of this page as completely as you can. If you qualified under multiple categories, use your ZIO roll number. Ask your centre supervisor if you do not know your roll number.

(g) At the beginning of each program, include a comment with your name, roll number and the problem number. For instance, if your name is Sachin Tendulkar, your roll number is 1600107 and the problem you are writing a C/C++/Java/Pascal program for is Problem 1, add a comment of the form

```
/*  Sachin Tendulkar, 1600107, Problem 1 */
```

at the top of your file.

This will help us if files accidentally get mixed up when being sent from the exam centres to the evaluation centre.

Contact details	
Roll Number:	
Name:	
Home Address:	PIN Code
Phone Number(s): (with STD code)	
Email address:	

Question 1 Special Sums

In this problem you are given two lists of N integers, $a_1, a_2, \ldots a_N$ and $b_1, b_2, \ldots b_N$. For any pair (i, j) with $i, j \in \{1, 2, \ldots, N\}$, we define the segment from i to j, written as $[i, j]$, to be $i, i+1, \ldots, j$ if $i \leq j$ and $i, i+1, \ldots, N, 1, 2, \ldots, j$ if $i > j$. Thus if $N = 5$ then the $[2, 4] = \{2, 3, 4\}$ and $[4, 2] = \{4, 5, 1, 2\}$.

With each segment $[i, j]$ we associate a special sum $SSum[i, j]$ as follows:

- $SSum[i, i] = a_i$.

- If $i \neq j$ then $SSum[i, j] = a_i + (\sum_{k \in [i,j], k \neq i, j} b_k) + a_j$. The positions i and j contribute a_i and a_j, respectively, to the sum while every other position k in $[i, j]$ contributes b_k.

Suppose $N = 5$ and that the two given sequences are as follows:

i	1	2	3	4	5
a_i	2	3	2	3	1
b_i	3	4	4	6	3

Then, $SSum[1, 1] = 2$, $SSum[2, 4] = 3 + 4 + 3 = 10$ and $SSum[4, 2] = 3 + 3 + 3 + 3 = 12$. Your aim is to compute the maximum value of $SSum[i, j]$ over all segments $[i, j]$. In this example you can verify that this value is 18 ($SSum[2, 1] = 18$).

Input format

- The first line contains a single positive integer N.

- This is followed by a line containing N integers giving the values of the a_is and this is followed by a line containing N integers giving the values of the b_is.

Output format

A single integer in a single line giving the maximum possible special segment sum.

Note The final value may not fit in a 32 bit integer. Use variables of an appropriate type to store and manipulate this value (`long long` in C/C++, `long` in Java).

Test Data

You may assume that $-10^9 \leq a_i, b_i \leq 10^9$.

Subtask 1 (10 Marks) $1 \leq N \leq 3000$.
Subtask 2 (20 Marks) $1 \leq N \leq 10^6$ and $a_i = b_i$ for all $1 \leq i \leq N$.
Subtask 3 (30 Marks) $3 \leq N \leq 10^6$. Further $a_1 = b_1 = a_N = b_N = -10^9$ and for each $1 < k < N$ we have $-999 \leq a_k, b_k \leq 999$.
Subtask 4 (40 Marks) $1 \leq N \leq 10^6$.

Example

Here is the sample input and output corresponding to the example above.

Sample input

```
5
2 3 2 3 1
3 4 4 6 3
```

Sample output

```
18
```

Time and memory limits

The time limit for this task is 3 seconds. The memory limit is 512 MB.

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Important

Indicate the location of the source code file for your solution to Question 1 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file:

Question 2   Periodic strings

A *string* is any nonempty sequence of 0s and 1s. Examples of strings are 00, 101, 111000, 1, 0, 01. The *length* of a string is the number of symbols in it. For example, the length of 111000 is 6. If u and v are strings, then uv is the string obtained by concatenating u and v. For example if $u = 110$ and $v = 0010$ then $uv = 1100010$.

A string w is *periodic* if there exists a string v such that $w = v^n = vv \cdots v$ (n times), for some $n \geq 2$. Note that in this case the length of v is strictly less than that of w. For example, 110110 is periodic, because it is vv for $v = 110$.

Given a positive integer N, find the number of strings of length N which are *not* periodic. Report the answer modulo M. The non-periodic strings of length 2 are 10 and 01. The non-periodic strings of length 3 are 001, 010, 011, 100, 101, and 110.

Input format

A single line, with two space-separated integers, N and M.

Output format

A single integer, the number of non-periodic strings of length N, modulo M.

Test Data

In all subtasks, $2 \leq M \leq 10^8$. The testdata is grouped into 4 subtasks.

Subtask 1 (10 marks) $1 \leq N \leq 4000$. N is the product of two distinct prime numbers.
Subtask 2 (20 marks) $1 \leq N \leq 4000$. N is a power of a prime number.
Subtask 3 (35 marks) $1 \leq N \leq 4000$.
Subtask 4 (35 marks) $1 \leq N \leq 150000$.

Example

Sample input

3 176

Sample output

6

Time and memory limits

The time limit for this task is 3 seconds. The memory limit is 512 MB.

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

3

Important

Indicate the location of the source code file for your solution to Question 2 in the box below.
If you fail to do this, your solution cannot be evaluated!

Source file: