Indian National Olympiad in Informatics, 2010

Time: 3 hours *24 January, 2010*

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 Pascal, C or C++ 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 `djgpp`, 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) Please fill out your contact details on the reverse of this page as completely as you can. Use your serial number as reported in the ZIO/ZCO results page as your roll number for INOI. If you qualified under multiple categories, use the ZIO serial number as your roll number. Ask your centre supervisor if you do not know your roll number.

(f) 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 107 and the problem you are writing a C/C++/Pascal program for is Problem 1, add a comment of the form

```
/*  Sachin Tendulkar, 107, 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.

. . . P.T.O.

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

Question 1 Vogon zoo

On the icy planet Zorg, the Vogons are putting together a zoo. One cage will house a collection of Kubudu dragons. Unlike the limited number of blood types found in other creatures, Kubudu dragons have a large variety of blood types. Each dragon's blood type is fixed when it is born and is given by a positive integer.

These blood types determine how the dragons interact with each other. Two dragons whose blood types are close to each other will quickly start fighting and eventually destroy each other. At any given time, there is a threshold K such that it is safe to put two Kubudu dragons in the same cage only if their blood types differ by K or more.

A batch of freshly hatched Kubudu dragons has arrived. Each dragon has had its blood tested and is tagged with its blood type. The Vogon zookeeper would like to determine the size of the largest collection of dragons from this batch that can safely be placed in a single cage. Given the dragons' fighting tendencies, this means that for each pair of dragons in the cage, their blood types should differ by at least K.

For instance, suppose that K is 3 and there are 12 Kubudu dragons whose blood types are 1, 5, 6, 1, 8, 3, 12, 2, 13, 7, 9 and 11. From this batch, the maximum number of dragons that can be placed safely in the same cage is 4—for example, the dragons with blood types 6, 12, 2 and 9.

You will be given the blood types of N Kubudu dragons and the threshold K. Your task is to compute the size of the largest group of dragons from this collection that can safely be placed in the same cage.

Input format

The first line of input has two space separated integers N and K, where N is the number of Kubudu dragons and K is the threshold below which they fight, as described above. The second line of input consists of N space separated integers, the blood types of the N dragons.

Output format

A single integer, the size of the largest collection of dragons that can be safely placed in the same cage.

Test data

In all cases, $1 \leq N \leq 10^6$. In 30% of the inputs, $1 \leq N \leq 5000$. The blood types of the dragons lie in the range 1 to 10^7.

Example

Sample input

```
12 3
1 5 6 1 8 3 12 2 13 7 9 11
```

Sample output

```
4
```

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 Twin robots

You have a board divided into $N \times N$ squares and two robots, R_2D_2 and C_3PO. Each square on the board has a number written on it.

R_2D_2 starts at the top left corner square of the board and C_3PO starts at the top right corner square of the board. Your aim is to move the robots so that R_2D_2 reaches the bottom right corner square and C_3PO reaches the bottom left corner square. You have a remote control with two buttons, blue and yellow, that control the robots as follows.

- Whenever you press the blue button, R_2D_2 moves one square to the right and C_3PO moves one square down.
- Whenever you press the yellow button, R_2D_2 moves one square down and C_3PO moves one square left.

The robots are not allowed to move off the board—if pushing a button would force either one of the robots to move off the board, the button push has no effect and both robots stay where they are.

For each robot, you compute a score which is the sum of the numbers on the squares that it visits along the path that it follows from its starting position to its final position. The combined score of the two robots is the sum of their individual scores. Your aim is to compute the maximum combined score that the two robots can achieve.

For example, suppose you have a 4×4 board as follows.

6	0	3	−1
7	4	2	4
−3	3	−2	8
13	10	−1	−4

In this example, if R_2D_2 follows the path marked by * and C_3PO follows the path marked by †, as shown below, their combined score is 56.

6^*	0	$3^\dagger$	$-1^\dagger$
7^*	$4^{*\dagger}$	$2^\dagger$	4
−3	$3^{*\dagger}$	-2^*	8^*
$13^\dagger$	$10^\dagger$	−1	-4^*

It can be verified that this is the best combined score that can be achieved for this board.

Input format

The first line of input is a single integer N, the size of the square board. This is followed by N lines, each containing N space separated integers. For $1 \leq i \leq N$, the integers on line $i{+}1$ of the input describe row i of the board.

Output format

A single integer, the maximum combined score that the two robots can achieve.

Test data

In all cases, $1 \leq N \leq 2500$. In 30% of the inputs, $1 \leq N \leq 40$. The score on each square is in the range -9999 to $+9999$.

Example

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

Sample input

```
4
6 0 3 -1
7 4 2 4
-3 3 -2 8
13 10 -1 -4
```

Sample output

```
56
```

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 2 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file: