



## Ngôn ngữ lập trình C++

### BÀI TẬP LẬP TRÌNH CƠ BẢN VÀ NÂNG CAO

#### A. Phần cơ bản



##### 1. Two Domino(1) – TwoDomino1.Cpp

Cho hai con Domino, Domino 1 có số chấm ở phần trên là  $A_1$ , số chấm ở phần dưới là  $B_1$ ; Domino 2 có số chấm ở phần trên là  $A_2$ , số chấm ở phần dưới là  $B_2$ .

**Yêu cầu:** Tính tổng số chấm của 2 Domino ở phần trên, tổng số chấm ở phần dưới.

**Dữ liệu** cho trong file **TwoDomino1.Inp** gồm:

- Dòng 1 ghi hai số nguyên dương  $A_1, B_1$  ( $1 \leq A_1, B_1 \leq 10$ ) lần lượt là số chấm ở phần trên và phần dưới của Domino 1.
- Dòng 2 ghi hai số nguyên dương  $A_2, B_2$  ( $1 \leq A_2, B_2 \leq 10$ ) lần lượt là số chấm ở phần trên và phần dưới của Domino 2.

**Kết quả** ghi ra file **TwoDomino1.Out** gồm 2 dòng:

- Dòng 1 ghi tổng số chấm phần trên của 2 Domino.
- Dòng 2 ghi tổng số chấm phần dưới của 2 Domino.

Ví dụ:

TwoDomino1.Inp	TwoDomino1.Out	Hình minh họa
2 5 2 6	4 11	



##### 2. Two Domino (2) – TwoDomino2.Cpp

Cho hai con Domino, Domino 1 có số chấm ở phần trên là  $A_1$ , số chấm ở phần dưới là  $B_1$ ; Domino 2 có số chấm ở phần trên là  $A_2$ , số chấm ở phần dưới là  $B_2$ .

Bạn có thể lật, hoặc không lật các con Domino. Khi lật một con Domino thì phần trên sẽ trở thành phần dưới, phần dưới sẽ trở thành phần trên.

Gọi  $X$  là tổng các chấm ở phần trên của 2 Domino.

Gọi  $Y$  là tổng các chấm ở phần dưới của 2 Domino.

**Yêu cầu:** Tính giá trị nhỏ nhất của  $|X - Y|$ .

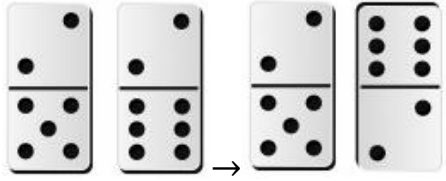
**Dữ liệu** cho trong file **TwoDomino2.Inp** gồm:

- Dòng 1 ghi hai số nguyên dương  $A_1, B_1$  ( $1 \leq A_1, B_1 \leq 10$ ) lần lượt là số chấm ở phần trên và phần dưới của Domino 1.
- Dòng 2 ghi hai số nguyên dương  $A_2, B_2$  ( $1 \leq A_2, B_2 \leq 10$ ) lần lượt là số chấm ở phần trên và phần dưới của Domino 2.

**Kết quả** ghi ra file **TwoDomino2.Out** là giá trị nhỏ nhất của  $|X - Y|$ .

Ví dụ:



TwoDomino2.Inp	TwoDomino2.Out	Hình minh họa
2 5 2 6	1	 Lật con Domino 2

**3. Three Domino – ThreeDomino.Cpp**

Cho ba con Domino, Domino thứ  $i$  ( $i = 1, 2, 3$ ) có số chấm ở phần trên là  $A_i$ , số chấm ở phần dưới là  $B_i$ .

Bạn có thể lật (hoặc không lật) nhiều nhất một con Domino. Khi lật một con Domino thì phần trên sẽ trở thành phần dưới, phần dưới sẽ trở thành phần trên.

Gọi  $X$  là tổng các chấm ở phần trên của 3 Domino.

Gọi  $Y$  là tổng các chấm ở phần dưới của 3 Domino.

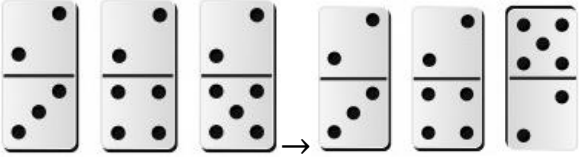
**Yêu cầu:** Tính giá trị nhỏ nhất của  $|X - Y|$ .

**Dữ liệu** cho trong file **ThreeDomino.Inp** gồm 3 dòng:

- Dòng thứ  $i$  ( $i = 1, 2, 3$ ) ghi hai số nguyên dương  $A_i, B_i$  ( $1 \leq A_i, B_i \leq 10$ ) lần lượt là số chấm ở phần trên và phần dưới của con Domino  $i$ .

**Kết quả** ghi ra file **ThreeDomino.Out** là giá trị nhỏ nhất của  $|X - Y|$ .

Ví dụ:

ThreeDomino.Inp	ThreeDomino.Out	Hình minh họa
2 3 2 4 2 5	0	 Lật con Domino thứ 3

**4. N Domino - NDomino.Cpp**

Cho  $N$  con Domino, Domino thứ  $i$  ( $i = 1, 2, \dots, N$ ) có số chấm ở phần trên là  $A_i$ , số chấm ở phần dưới là  $B_i$ .

Bạn có thể lật (hoặc không lật) nhiều nhất một con Domino. Khi lật một con Domino thì phần trên sẽ trở thành phần dưới, phần dưới sẽ trở thành phần trên.

Gọi  $X$  là tổng các chấm ở phần trên của  $N$  Domino.

Gọi  $Y$  là tổng các chấm ở phần dưới của  $N$  Domino.

**Yêu cầu:** Tính giá trị nhỏ nhất của  $|X - Y|$ .

**Dữ liệu** cho trong file **NDomino.Inp** gồm:

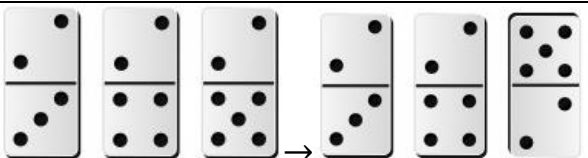
- Dòng đầu ghi số nguyên dương  $N$  ( $N \leq 10^5$ ).



- $N$  dòng sau, dòng thứ  $i$  ( $i = 1, 2, \dots, N$ ) ghi hai số nguyên dương  $A_i, B_i$  ( $1 \leq A_i, B_i \leq 100$ ) lần lượt là số chấm ở phần trên và phần dưới của con Domino  $i$ .

**Kết quả** ghi ra file **NDomino.Out** là giá trị nhỏ nhất của  $|X - Y|$ .

Ví dụ:

NDomino.Inp	NDomino.Out	Hình minh họa
3 2 3 2 4 2 5	0	 <p>Lật con Domino thứ 3</p>

## A. Phân Nâng cao



### 1. Số cân bằng

Số tự nhiên  $n$  được gọi là số cân bằng nếu tất cả các chữ số của  $n$  đều bằng nhau. Ví dụ: 0, 111, 1, 2, 22 là các số cân bằng.

Cho dãy số nguyên dương  $a_1, a_2, \dots, a_n$ .

**Yêu cầu:** Đếm xem có bao nhiêu số hạng trong dãy là số cân bằng.

**Dữ liệu:** Vào từ file văn bản NUMBALANCE.INP gồm:

- Dòng đầu tiên chứa số nguyên  $n$  ( $n \leq 10^6$ ).
- Dòng thứ 2 chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $a_i \leq 10^9$ ).

**Kết quả:** Đưa ra file văn bản NUMBALANCE.OUT là số các số hạng là số cân bằng.

Ví dụ:

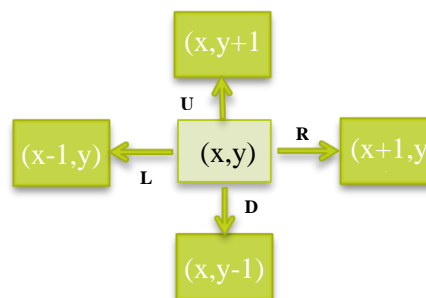
NUMBALANCE.INP	NUMBALANCE.OUT
5 1 12 22 33 31	3



### 2. Robot (1)

Trên lưới nguyên Oxy có đặt một con robot tại tọa độ  $(x_0, y_0)$ . Có bốn lệnh điều khiển robot như sau:

- Lệnh U: robot di chuyển sang ô kề trên
- Lệnh D: robot di chuyển sang ô kề dưới
- Lệnh L: robot di chuyển sang ô kề trái
- Lệnh R: robot di chuyển sang ô kề phải



**Yêu cầu:**



- Cho biết dãy các lệnh điều khiển robot, tìm tọa độ robot khi thực hiện xong dãy lệnh đã cho.
- Cho biết tọa độ các điểm lần lượt mà robot đi qua, hãy tìm dãy các lệnh điều khiển.

**Dữ liệu** cho trong file ROBOTMV.INP gồm:

- Dòng đầu ghi tọa độ  $x_0, y_0$  ban đầu của robot ( $|x_0|, |y_0| \leq 10^6$ ).
- Dòng sau ghi một xâu gồm các kí tự thuộc  $\{U, D, R, L\}$  mô tả các lệnh điều khiển, (độ dài xâu không quá  $10^5$ ).
- Dòng thứ ba ghi số  $n$  ( $n \leq 10^5$ ) là số tọa độ mà robot lần lượt đi qua.
- $n$  dòng cuối, mỗi dòng ghi hai số nguyên tương ứng là tọa độ mà robot lần lượt đi qua.

**Kết quả** ghi ra file ROBOTMV.OUT gồm:

- Dòng đầu ghi hai số  $x, y$  là tọa độ của robot sau khi di chuyển theo dãy lệnh đã cho.
- Dòng thứ hai ghi xâu gồm  $n-1$  kí tự thuộc  $\{L, R, U, D\}$  mô tả dãy lệnh để robot lần lượt đi qua  $n$  đỉnh đã cho (dữ liệu luôn đảm bảo có nghiệm).

Ví dụ:

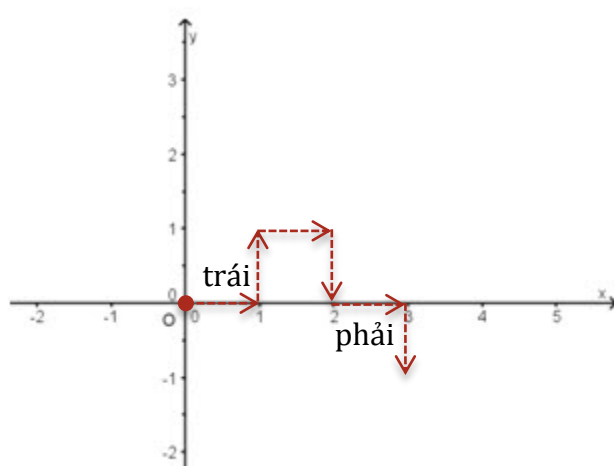
ROBOTMV.INP	ROBOTMV.OUT
0 0	2 0
RR	URLD
5	
4 4	
4 5	
5 5	
4 5	
4 4	



### 3 Robot (2)

Cho lưới nguyên Oxy. Điểm nguyên  $(x_1, y_1)$  và điểm nguyên  $(x_2, y_2)$  được gọi là kề nhau nếu thỏa điều kiện  $|x_1 - x_2| + |y_1 - y_2| = 1$ . Một robot ban đầu đứng tại gốc tọa độ. Ở mỗi bước, robot sẽ di chuyển sang một điểm nguyên kề với vị trí hiện tại. Từ bước di chuyển thứ hai trở đi, robot có thể đi tiếp theo hướng cũ, rẽ sang trái, rẽ sang phải, hay trở lại vị trí trước đó.

Ví dụ từ ô  $(0, 0)$ , robot đi đến  $(1, 0)$ , rẽ trái sang ô  $(1, 1)$ , rẽ phải sang ô  $(2, 1)$ , rẽ phải sang ô  $(2, 0)$ , rẽ trái sang  $(3, 0)$  cuối cùng rẽ phải sang ô  $(3, -1)$ .



**Yêu cầu:** Cho tọa độ các điểm nguyên mà robot lần lượt đã đi qua. Hãy đếm xem robot đã rẽ phải bao nhiêu lần, rẽ trái bao nhiêu lần?

**Dữ liệu** cho trong file ROBOT.INP gồm:

- Dòng đầu tiên chứa 1 số nguyên dương  $n$  (với  $2 \leq n \leq 10^6$ ) là tổng số điểm nguyên mà robot đã đi qua (kể cả vị trí xuất phát là gốc tọa độ),
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo ( $1 \leq i \leq n$ ) chứa 2 số nguyên  $x_i$  và  $y_i$  là tọa độ điểm nguyên mà robot đã đi qua. Các số trên cùng một dòng được ghi cách nhau bởi 1 khoảng trắng.

**Kết quả** ghi ra file ROBOT.OUT gồm một dòng chứa hai số nguyên là số lần robot đã rẽ phải và số lần robot đã trái.

*Ví dụ:*

ROBOT.INP	ROBOT.OUT
7 0 0 1 0 1 1 2 1 2 0 3 0 3 -1	3 2



## 4. GCD trên dãy thay đổi một số hạng

Cho dãy số nguyên  $A_1, A_2, \dots, A_N$ . Bạn hãy chọn một số hạng trong dãy và thay đổi số hạng này bằng một giá trị nguyên thuộc  $[1; 10^9]$ . Giá trị số hạng được chọn có thể không thay đổi.

**Yêu cầu:** Hãy tìm cách chọn một số hạng và thay đổi giá trị này để ước chung lớn nhất của dãy  $A_1, A_2, \dots, A_N$  có giá trị lớn nhất.

**Dữ liệu** cho trong file **ChangeOneGCD.Inp** gồm:

- Dòng đầu ghi số nguyên dương  $N$ .



- Dòng hai ghi  $N$  số nguyên dương  $A_1, A_2, \dots, A_N$ .

**Kết quả** ghi ra file **ChangeOneGCD.Out** là giá trị lớn nhất có thể của ước chung lớn nhất của dãy  $A_1, A_2, \dots, A_N$ .

Ví dụ:

ChangeOneGCD.Inp	ChangeOneGCD.Out	Giải thích
3 2 5 10	5	Thay 2 bằng 5

**Giới hạn:**

- $2 \leq N \leq 10^5$ ;
- $1 \leq A_i \leq 10^9$ .



## 5. Số bạn bè - NumFriend.Cpp

Cho số nguyên dương  $n$ , ta gọi  $S(n)$  là tổng các ước dương của  $n$  và khác  $n$ . Ví dụ:  $n = 4$ ,  $S(n) = 1 + 2 = 3$ .

Cặp số nguyên dương  $a$  và  $b$  được gọi là cặp số bạn bè nếu  $S(a) = b$  và  $S(b) = a$  với  $a \neq b$ .

**Yêu cầu:** Cho hai số nguyên dương  $L$  và  $R$ . Tìm số cặp  $(a, b)$  là số bạn bè với  $L \leq a < b \leq R$ .

**Dữ liệu** cho trong file **NumFriend.Inp** gồm 2 số nguyên dương  $L$  và  $R$ .

**Kết quả** ghi ra file **NumFriend.Out** là số cặp  $(a, b)$  là số bạn bè với  $L \leq a < b \leq R$ .

Ví dụ:

NumFriend.Inp	NumFriend.Out
220 285	1

**Giới hạn:**

- Có 50% test ứng với  $1 \leq L < R \leq 1000$  ;
- Có 25% test ứng với  $1 \leq L < R \leq 100.000$  ;
- Có 25% test ứng với  $1 \leq L < R \leq 5.000.000$  ;