



Ngôn ngữ lập trình C++

BÀI TẬP CƠ BẢN VÀ NÂNG CAO

A. PHẦN CƠ BẢN

Bài tập 1. Tráo đổi phần tử giữa hai mảng – SwapArr.Cpp

Cho hai dãy số A_1, A_2, \dots, A_N và B_1, B_2, \dots, B_N .

Với mọi chỉ số i (với $i = 1, 2, 3, \dots, N$) nếu $A_i > B_i$ thì tráo đổi giá trị của A_i và B_i cho nhau (tức là A_i nhận giá trị của B_i , B_i lại nhận giá trị của A_i ban đầu).

Yêu cầu: Đưa ra dãy A_1, A_2, \dots, A_N và B_1, B_2, \dots, B_N sau khi tráo đổi.

Dữ liệu: Nhập số nguyên dương N ($N \leq 1000$).

- Dòng 2 nhập N số nguyên dương A_1, A_2, \dots, A_N ($A_i \leq 1000$).
- Dòng 3 nhập N số nguyên dương B_1, B_2, \dots, B_N ($B_i \leq 1000$).

Kết quả: Đưa ra:

- Dòng 1: Dãy A_1, A_2, \dots, A_N ;
- Dòng 2: Dãy B_1, B_2, \dots, B_N .

Ví dụ:

Input	Output
5	1 2 2 5 6
1 3 2 5 6	1 3 7 8 8
1 2 7 8 8	

Bài tập 2. Tính tổng phân số (1) - SumFract1.Cpp

Cho số nguyên dương n . Tính tổng $S = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \dots + \frac{2n-1}{2n}$

Đưa ra giá trị của S với độ chính xác 6 chữ số thập phân.

Dữ liệu: Nhập số nguyên n ($3 \leq n \leq 100000$).

Kết quả: Đưa ra giá trị của S với độ chính xác 6 chữ số thập phân.

Ví dụ:

Input	Output
5	3.858333

Gợi ý khai báo dữ liệu:

- Biến chứa tổng: **double S;**
- Câu lệnh đưa ra số thực với 6 chữ số thập phân:

```
cout<<setprecision(6)<<fixed<<s;
```

Bài tập 3. Tính tổng phân số (2) – SumFract2.Cpp

Cho số nguyên dương n . Tính tổng $S = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \frac{1}{1+2+3+4} + \dots + \frac{1}{1+2+3+\dots+n}$



Đưa ra giá trị của S với độ chính xác 6 chữ số thập phân.

Dữ liệu: Nhập số nguyên N ($3 \leq n \leq 100000$).

Kết quả: Đưa ra giá trị của S với độ chính xác 6 chữ số thập phân.

Ví dụ:

Input	Output
5	1.666667

Bài tập 4. Lập trình tính số PI – PIValue.Cpp

Trong toán học số PI (π) là một số vô hạn không tuần hoàn. Tức là nó có vô hạn số thập phân mà các số thập phân này không có sự tuần hoàn. Trong các công thức tính chu vi, diện tích ta thường sử dụng giá trị số PI gần đúng chẳng hạn bằng 3.14.

Có nhiều công thức để tính giá trị gần đúng số PI, sau đây ta xét công thức:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \dots + (-1)^n \frac{4}{2n+1}$$

Cho giá trị n ($n \leq 10^6$), hãy đưa ra giá trị gần đúng của PI.

Dữ liệu: Nhập giá trị của n .

Kết quả: Đưa ra giá trị của PI có 6 chữ số thập phân.

Ví dụ:

Input	Output
1000	3.142592

B. PHẦN NÂNG CAO



1. Kẽ 4 đoạn

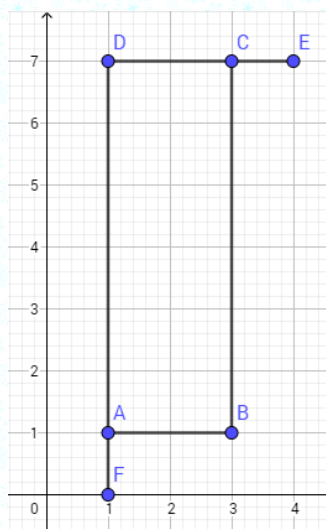
Trong mặt phẳng tọa độ Oxy, hãy kẻ 4 đoạn thẳng có độ dài a, b, c, d sao cho những điểm giao của 4 đường kẻ này tạo thành một hình chữ nhật có diện tích lớn nhất.

Dữ liệu cho trong file **FourSeg.Inp** gồm 4 số nguyên dương a, b, c, d ($a, b, c, d \leq 10^6$).

Kết quả ghi ra file **FourSeg.Out** là diện tích lớn nhất của hình chữ nhật có thể tạo được.

Ví dụ:

FourSeg.Inp	FourSeg.Out
3 2 6 7	12

**2. Cặp có tổng bằng K**

Xét dãy số nguyên A gồm N số hạng A_1, A_2, \dots, A_N . Ta gọi $S(A)$ là số các cặp chỉ số (i, j) sao cho: $1 \leq i < j \leq N$ và $A_i + A_j = K$ với K là giá trị cho trước.

Ví dụ: Dãy $A: [1, 2, 3, 4, 7], K = 5$ thì $S(A) = 2$.

Yêu cầu: Cho dãy số nguyên B_1, B_2, \dots, B_N và K , hãy tìm cách chia N số hạng này thành 2 nhóm để tạo thành 2 dãy X và Y sao cho $S(X) + S(Y)$ đạt giá trị nhỏ nhất.

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

- Dòng đầu ghi hai số nguyên N và K .
- Dòng thứ 2 ghi N số nguyên A_1, A_2, \dots, A_N .

Kết quả ghi ra file **SumPK.Out** là tổng $S(X) + S(Y)$ đạt giá trị nhỏ nhất.

Ví dụ:

SumPK.Inp	SumPK.Out	Giải thích
5 5 1 2 3 4 7	0	Có thể chia làm 2 dãy: $X = [1, 2, 7], Y = [3, 4]$

Giới hạn:

- $|A_i| \leq 10^{12}$;
- $2 \leq N \leq 10^5$;
- $|K| \leq 10^{12}$.

**3. Liệt kê hoán vị**

Cho số nguyên dương N ($3 \leq N \leq 20$). Hãy liệt kê tất cả các hoán vị của N số tự nhiên $\{1, 2, 3, 4, \dots, N\}$.

Dữ liệu cho trong file **Permutation.Inp** gồm số nguyên dương N .

Kết quả ghi ra file **Permutation.Out** là các hoán vị của N số tự nhiên $\{1, 2, 3, 4, \dots, N\}$. Các hoán vị được đưa ra theo thứ tự từ điển, mỗi hoán vị ghi trên một dòng, các số trong hoán vị được ghi cách nhau bởi dấu cách. Nếu số hoán vị nhiều hơn 2021 thì chỉ ghi 2021 hoán vị đầu tiên.

Ví dụ:



Permutation.Inp	Permutation.Out
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1



4. Xếp số

Có K chữ số D_1, D_2, \dots, D_K đôi một khác nhau và lớn hơn 0. Hãy dùng K chữ số này để ghép thành các số X thỏa mãn:

- Tất cả các chữ số trong K chữ số đều có trong X .
- Không có chữ số nào xuất hiện quá 2 lần.
- Nhiều nhất có 1 chữ số xuất hiện đúng 2 lần.

Yêu cầu: Đưa ra tất cả các số X tạo được theo thứ tự tăng dần.

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

- Dòng đầu ghi số nguyên dương K ($1 \leq K \leq 5$).
- Dòng 2 ghi K chữ số D_1, D_2, \dots, D_K .

Kết quả ghi ra file **PuzzleNum.Out** là các số X tạo được, các số đưa ra theo thứ tự tăng dần, mỗi số ghi trên một dòng.

Ví dụ:

PuzzleNum.Inp	PuzzleNum.Out
2	37
3 7	73
	337
	373
	377
	733
	737
	773



5. Tráo đổi nguyên tố

Cho số nguyên dương X ($X \leq 10^8$). Hãy tìm cách tráo đổi các chữ số của X để nhận được một số Y là số nguyên tố.

Yêu cầu: Hãy đưa ra tất cả các giá trị của Y có thể nhận được.

Dữ liệu cho trong file **SwapPrime.Inp** gồm một số nguyên dương X .

Kết quả ghi trong file **SwapPrime.Out** là các giá trị của Y , mỗi số ghi trên một dòng theo thứ tự từ nhỏ đến lớn. Nếu không có giá trị nào của Y , tức là không thể tráo đổi các chữ số của X để nhận được số nguyên tố thì ghi “Hello2021”.

Ví dụ:



SwapPrime.Inp	SwapPrime.Out
224	Hello2021
13	13 31