Problems in Elementary Computer Science – Bài Tập Tin Học Sơ Cấp

Nguyễn Quản Bá Hồng*

Ngày 11 tháng 5 năm 2023

Tóm tắt nội dung

1 bộ sưu tập các bài toán chọn lọc từ cơ bản đến nâng cao cho Tin học sơ cấp. Phiên bản mới nhất của tài liệu này được lưu trữ ở link sau: GitHub/NQBH/hobby/elementary computer science/problem[1].

Mục lục

1 Notes on Commands

1.1 Notes on C/C++ commands

1.2 Notes on Pascal commands

1.3 Notes on Python commands

1. Để sử dụng các hàm toán học trong Python, cần import thư viện `math` vào chương trình: `from math import *`

*Independent Researcher, Ben Tre City, Vietnam
e-mail: nguyenquanbahong@gmail.com; website: https://nqbh.github.io.

[1]URL: https://github.com/NQBH/hobby/blob/master/elementary_computer_science/problem/NQBH_elementary_computer_science_problem.pdf.

2. Để mở file dữ liệu vào `prob.inp` chỉ để đọc dữ liệu & mở file dữ liệu ra `prob.out` để thay đổi dữ liệu trong file: `file = open("prob.inp")` & `file2 = open("prob.out", "w")`.

3. Sắp xếp trong Python có thể thực hiện 1 cách đơn giản nhờ phương thức `sort()` hoặc `sorted()`. Cú pháp:

```
list.sort(reverse = True|False, key = myFunc)
list.sorted(reverse = True|False, key = myFunc)
```

4. Trong Python, để lấy giá trị ngẫu nhiên, sử dụng phương thức `ranint(a, b)` với `a, b` là giới hạn của giá trị cần lấy ngẫu nhiên.

2 Problems in Elementary Mathematics – Bài Toán Tin Học Trong Toán Học Sơ Cấp

Bài toán 1 (Even vs. odd). *Viết thuật toán & các chương trình bằng các ngôn ngữ lập trình* Pascal, Python, C/C++ *để xét tính chẵn lẻ của* $n \in \mathbb{Z}$ *được nhập từ bàn phím.*

- Pascal script: GitHub/NQBH/hobby/elementary computer science/Pascal/even vs. odd.
- Python script: GitHub/NQBH/hobby/elementary computer science/Python/even vs. odd.

Bài toán 2 (Divisible by). *Viết thuật toán & các chương trình bằng các ngôn ngữ lập trình* Pascal, Python, C/C++ *để kiểm tra liệu* $a \vdots b$ *hay không, với* $a, b \in \mathbb{Z}$ *được nhập từ bàn phím.*

- Pascal script: GitHub/NQBH/hobby/elementary computer science/Pascal/divisible by.
- Python script: GitHub/NQBH/hobby/elementary computer science/Python/divisible by.

Bài toán 3 (Triangle). *Viết thuật toán & các chương trình bằng các ngôn ngữ lập trình* Pascal, Python, C/C++ *để liệu* a, b, c *có phải là độ dài của: (a) 1 tam giác. (b) 1 tam giác nhọn. (c) 1 tam giác vuông. (d) 1 tam giác tù.*

- Python script: GitHub/NQBH/hobby/elementary computer science/Python/triangle.

Bài toán 4 (Polynomial equation). *Viết thuật toán & các chương trình bằng các ngôn ngữ lập trình* Pascal, Python, C/C++ *để giải phương trình bậc nhất, bậc 2, bậc 3, & bậc 4 với các hệ số thực được nhập từ bàn phím.*

Bài toán 5 (Fibonacci sequence). *Viết thuật toán & các chương trình bằng các ngôn ngữ lập trình* Pascal, Python, C/C++ *để xuất ra màn hình, với* $n \in \mathbb{N}$ *được nhập từ bàn phím: (a) Số Fibonacci thứ* n*. (b)* n *số Fibonacci đầu tiên.*

Bài toán 6 (1st n square roots). *Viết chương trình* Pascal, C/C++, Python *xuất ra căn bậc 2 của* n *số tự nhiên đầu tiên với* $n \in \mathbb{N}^\star$ *được nhập từ bàn phím.*

Bài toán 7 (Số chính phương – Square number). *Viết chương trình* Pascal, C/C++, Python *để kiểm tra 1 số* $n \in \mathbb{N}^\star$ *được nhập từ bàn phím có phải là số chính phương hay không.*

Bài toán 8 (1st n cube roots). *Viết chương trình* Pascal, C/C++, Python *xuất ra căn bậc 3 của* n *số tự nhiên đầu tiên với* $n \in \mathbb{N}^\star$ *được nhập từ bàn phím.*

Bài toán 9. *Viết chương trình* Pascal, C/C++, Python *để kiểm tra 1 số* $n \in \mathbb{N}^\star$ *được nhập từ bàn phím có phải là lập phương của 1 số tự nhiên hay không.*

Bài toán 10 (1st n nth roots). *Viết chương trình* Pascal, C/C++, Python *xuất ra căn bậc* n *của* m *số tự nhiên đầu tiên với* $m, n \in \mathbb{N}^\star$ *được nhập từ bàn phím.*

Bài toán 11. *Viết chương trình* Pascal, C/C++, Python *để kiểm tra 1 số* m *được nhập từ bàn phím có phải là lũy thừa bậc* n *của 1 số tự nhiên hay không với* $m, n \in \mathbb{N}^\star$ *được nhập từ bàn phím.*

3 Algebraic Expression – Biểu Thức Đại Số

Bài toán 12 ([Vie21], 1., p. 15, Vũng Tàu 2020). *Cho* $a, b, c \in \mathbb{N}^\star$. Yêu cầu: *Tính giá trị của biểu thức* $S = \dfrac{a^2 + b^2 + c^2}{abc} + \sqrt{abc}$.

- Dữ liệu vào: *File* `root.inp` *chứa 3 số nguyên dương* a, b, c*. Mỗi số trên 1 dòng.*
- Kết quả: *Ghi vào File* `root.out` *kết quả* S *tính được (làm tròn lấy 2 chữ số sau phần thập phân). E.g.:*

root.inp	root.out
2 1 2	4.25

Python script: GitHub/NQBH/hobby/elementary computer science/Python/root.py[2]. Input: root.inp. Output: root.out.

```
from math import *
file_in = open("root.inp")
file_out = open("root.out", "w")
a = file_in.readline()
b = file_in.readline()
c = file_in.readline()
a = int(a)
b = int(b)
c = int(c)
S = (a**a + b**b + c**c)/(a*b*c) + sqrt(a*b*c)
S = str(round(S,2))
file_out.write(S)
file_in.close()
file_out.close()
```

Lưu ý 1. *Tương tự, ta có thể tính hầu như bất kỳ hàm số $f(a, b, c)$ 3 biến a, b, c với f là 1 hàm số có thể viết được nhờ thư viện* `math` *của Python. Tổng quát hơn, ta có thể tính bất kỳ hàm số nhiều biến $f(x_1, x_2, \dots, x_n)$ với x_i, $i = 1, 2, \dots, n$, $n \in \mathbb{N}^\star$ là các biến, với f là 1 hàm số có thể viết được nhờ thư viện* `math` *của Python.*

Bài toán 13 ([Vie21], 2., p. 19, Bắc Giang 2020)**.** *Nhà An có 1 trang trại rộng lớn. Do sở thích của An nên bố An chỉ nuôi gà & chó. 1 hôm bố An đố con gái nhà mình nuôi bao nhiêu gà, bao nhiêu chó? Bố An cho biết nhà có tổng số gà & chó là x con. Do số lượng nhiều & khó đếm từng loại nên An chỉ đếm được tổng số chân của gà & chó là y chân. Giúp An trả lời câu đố.*

- Dữ liệu vào: *Đọc từ file văn bản* `toanco.inp` *gồm 2 số nguyên dương x, y trên 1 dòng. 2 số cách nhau 1 khoảng trống ($x \leq 10^5$, $y \leq 4 \cdot 10^5$).*
- Kết quả: *Ghi ra file văn bản* `toanco.out` *gồm 2 số tương ứng là số gà & số chó tìm được. 2 số cách nhau 1 khoảng trống. Giả sử bài toán luôn có nghiệm.*

toanco.inp	toanco.out
36 100	22 14

Python script: GitHub/NQBH/hobby/elementary computer science/Python/toanco.py[3]. Input: toanco.inp. Output: toanco.out.

```
file_in = open("toanco.inp")
file_out = open("toanco.out", "w")
s = file_in.readline()
s = s.split()
x = int(s[0])
y = int(s[1])
a = int(2*x - y/2)
b = int(y/2 - x)
file_out.write(str(a) + " " + str(b))
file_in.close()
file_out.close()
```

Bài toán 14 ([Vie21], 4., p. 26, Quãng Ngãi 2020, Lãi suất– Interest rate)**.** *1 người gửi tiền vào ngân hàng có kỳ hạn là c tháng với lãi suất mỗi tháng là $k\%$, số tiền gửi ban đầu là A (đơn vị triệu đồng).*

- Yêu cầu: *Tính số tiền người đó nhận được sau t tháng. Biết tiền lãi mỗi tháng được cộng dồn vào tiền gốc, nếu nhận tiền trước kỳ hạn thì số tiền được tính với lãi suất không kỳ hạn là $h\%$ của số tiền ban đầu A nhân với số tháng đã gửi. Trong trường hợp rút tiền sau kỳ hạn thì số tháng sau kỳ hạn sẽ được tính với lãi suất không kỳ hạn là $h\%$ so với số tiền thu được đã qua kỳ hạn.*
- Dữ liệu vào: *Tệp văn bản* `bl2.inp` *ghi 5 số kỳ hạn c (nếu $c = 0$ là gửi không kỳ hạn), thời gian gửi t, số tiền ban đầu A, lãi suất có kỳ hạn k, lãi suất không kỳ hạn h, các số cách nhau 1 ký tự trắng.*
- Dữ liệu ra: *Tệp văn bản* `bl2.out` *ghi 1 số là số tiền nhận được (làm tròn đến 1 số lẻ sau dấu chấm thập phân). E.g.:*

bl2.inp	bl2.out
12 13 100 1.0 0.2	112.9
0 10 100 1.0 0.2	102.0

[2] URL: https://github.com/NQBH/hobby/blob/master/elementary_computer_science/Python/root.py.
[3] URL: https://github.com/NQBH/hobby/blob/master/elementary_computer_science/Python/toanco.py.

4 Number Theory – Số Học

Bài toán 15 ([Vie21], 3., p. 20, Yên Bái 2020, Tổng nguyên tố)**.** *Viết chương trình nhập vào 2 số nguyên $a, b \in \mathbb{Z}$, $0 < a < b$. (a) Tìm & tính tổng các số nguyên tố của dãy số từ a đến b. (b) Xuất ra màn hình các số chia hết cho 5 của dãy số từ a đến b. (c)* (Bội của $n \in \mathbb{N}^\star$) *Xuất ra màn hình các số chia hết cho n của dãy số từ a đến b với $n \in \mathbb{N}^\star$ được nhập từ bàn phím. E.g., nhập $a = 6$, $b = 22$. Kết quả tổng các số nguyên tố trong dãy số từ 6 đến 22: $7 + 11 + 13 + 17 + 19 = 67$. Các số chia hết cho 5 của dãy số từ 6 đến 22: $10, 15, 20$.*

Bài toán 16 ([Vie21], 4., p. 22, Hải Dương 2020, Số mạnh mẽ)**.** Số mạnh mẽ *là số khi nó chia hết cho 1 số nguyên tố thì cũng chia hết cho cả bình phương của số nguyên tố đó, i.e., $a \in \mathbb{N}^\star$ là số mạnh mẽ $\Leftrightarrow (a \vdots p \Rightarrow a \vdots p^2, \forall p$: prime). E.g., 25 là số mạnh mẽ, vì nó chia hết cho số nguyên tố 5 & chia hết cho cả $5^2 = 25$. Viết chương trình liệt kê các số mạnh mẽ không vượt quá 1000.*

See, e.g., Wikipedia/powerful number, MathWorld/powerful number.

Bài toán 17 ([Vie21], 5., p. 23, Việt Nam 2020, Bội chính phương)**.** *Cho 1 dãy số A có n phần tử. Tìm số nguyên dương P nhỏ nhất thỏa mãn: a là số chính phương & a chia hết cho tất cả các phần tử của dãy số A.*

- Yêu cầu: *In ra phần dư của phép chia khi chia a cho $10^9 + 7$.*
- Dữ liệu vào: *Vào từ thiết bị theo khuôn dạng sau: Dòng đầu tiên chứa số nguyên dương n là số lượng phần tử của dãy số. Dòng tiếp theo chứa n số nguyên dương là các phần tử của dãy A. Các số trên 1 dòng được ghi cách nhau bởi dấu cách.*
- Kết quả: *Ghi ra thiết bị ra gồm 1 số nguyên duy nhất là kết quả của bài toán. E.g.:*

Dữ liệu vào	Dữ liệu ra
3 2 1 3	36

Bài toán 18 ([Vie21], 1., p. 25, Hải Dương 2020, Số hạnh phúc & số buồn bã – Happy- & sad numbers)**.** *Với 1 số nguyên dương bất kỳ, thay thế số đó bằng tổng bình phương các chữ số của nó & cứ lặp lại quá trình đó sẽ có các trường hợp sau xảy ra: Kết thúc bằng 1 – ta gọi số đó là* số hạnh phúc/happy number. *Kết thúc bằng 0 – ta gọi số đó là* số buồn bã/sad number. *Lặp lại vô hạn lần – số đó không hạnh phúc cũng không buồn bã. E.g., số 44: lần 1: $4^2 + 4^2 = 32$, lần 2: $3^2 + 2^2 = 13$, lần 3: $1^2 + 3^2 = 10$, lần 4: $1^2 + 0^2 = 1$, nên 44 là số hạnh phúc. Viết chương trình để kiểm tra xem ngày sinh của 1 người bất kỳ có phải là số hạnh phúc không?*

Bài toán 19 ([Vie21], 2., p. 25, Gia Lai 2019, Phân số tối giản – Irreducible fraction)**.** *1 chuỗi được gọi là có dạng phân số nếu nó có dạng* `'tử_số/mẫu_số'`*. Viết chương trình nhập vào chuỗi có dạng phân số, sau đó xuất ra dạng tối giản của phân số đó. E.g., Chuỗi* `'12/15'` *biểu diễn cho phân số. Dạng tối giản của phân số đó là* `'3/5'`*.*

Bài toán 20 (Tổng tất cả, tổng phần tử chẵn, lẻ, bình phương, lập phương, lũy thừa bậc n, căn bậc 2, 3, & căn bậc n, nghịch đảo, nghịch đảo bình phương, nghịch đảo lập phương, nghịch đảo lũy thừa bậc n, nghịch đảo căn bậc 2, 3, & nghịch đảo căn bậc n – Sums of all, odds, evens, squares, cubes, nth powers, square roots, cube roots, nth roots, reciprocals of square, of cubes, of nth powers, of square roots, of cube roots, of nth roots)**.** *Cho 1 dãy gồm n số nguyên: $(a_i)_{i=1}^m = a_1, a_2, \ldots, a_m$, $m \in \mathbb{N}^\star$, $a_i \in \mathbb{Z}$, $\forall i = 1, 2, \ldots, m$, mỗi số có giá trị không vượt quá 10^9.*

- Yêu cầu: *Tính tổng S tất cả các phần tử, tổng S_{even} các số chẵn, tổng S_{odd} các số lẻ, tổng S_{sqr} bình phương, tổng $S_{\text{sqr,even}}$ bình phương các số chẵn, tổng $S_{\text{sqr,odd}}$ bình phương các số lẻ, tổng S_{cb} lập phương, tổng $S_{\text{cb,even}}$ lập phương các số chẵn, tổng $S_{\text{cb,odd}}$ lập phương các số lẻ, tổng $S_{\text{pwr},n}$ lũy thừa bậc n, tổng $S_{\text{pwr,even},n}$ lũy thừa bậc n các số chẵn, tổng $S_{\text{pwr,odd},n}$ lũy thừa bậc n các số lẻ, tổng S_{sqrt} căn bậc 2, tổng $S_{\text{sqrt,even}}$ căn bậc 2 các số chẵn, tổng $S_{\text{sqrt,odd}}$ căn bậc 2 các số lẻ, tổng S_{cbrt} căn bậc 3, tổng $S_{\text{cbrt,even}}$ căn bậc 3 các số chẵn, tổng $S_{\text{cbrt,odd}}$ căn bậc 3 các số lẻ, tổng $S_{\text{rt},n}$ căn bậc n của các số, tổng $S_{\text{rt,even},n}$ căn bậc n của các số chẵn, tổng $S_{\text{rt,odd},n}$ căn bậc n của các số lẻ trong dãy $(a_i)_{i=1}^m$.*
- Dữ liệu: *Dòng đầu tiên chứa $m \in \mathbb{N}^\star$, $1 \le m \le 10^9$. Dòng thứ 2 chứa $n \in \mathbb{N}^\star$. m dòng tiếp theo, dòng thứ $i + 2$ chứa a_i, $\forall i = 1, 2, \ldots, m - 1$.*

Giải. Công thức toán học tính các tổng:

$$S \coloneqq \sum_{i=1}^{m} a_i = a_1 + a_2 + \cdots + a_m,\ S_{\text{even}} \coloneqq \sum_{i=1,\, 2|a_i}^{m} a_i,\ S_{\text{odd}} \coloneqq \sum_{i=1,\, 2\nmid a_i}^{m} a_i,$$

$$S_{\text{sqr}} \coloneqq \sum_{i=1}^{m} a_i^2 = a_1^2 + a_2^2 + \cdots + a_m^2,\ S_{\text{sqr,even}} \coloneqq \sum_{i=1,\, 2|a_i}^{m} a_i^2,\ S_{\text{sqr,odd}} \coloneqq \sum_{i=1,\, 2\nmid a_i}^{m} a_i^2,$$

$$S_{\text{cb}} \coloneqq \sum_{i=1}^{m} a_i^3 = a_1^3 + a_2^3 + \cdots + a_m^3,\ S_{\text{cb,even}} \coloneqq \sum_{i=1,\, 2|a_i}^{m} a_i^3,\ S_{\text{odd}} \coloneqq \sum_{i=1,\, 2\nmid a_i}^{m} a_i^3,$$

$$S_{\text{pwr},n} \coloneqq \sum_{i=1}^{m} a_i^n = a_1^n + a_2^n + \cdots + a_m^n,\ S_{\text{pwr,even},n} \coloneqq \sum_{i=1,\, 2|a_i}^{m} a_i^n,\ S_{\text{pwr,odd},n} \coloneqq \sum_{i=1,\, 2\nmid a_i}^{m} a_i^n,\ \forall n \in \mathbb{N}^\star,$$

$$S_{\text{sqrt}} \coloneqq \sum_{i=1}^{m} \sqrt{a_i} = \sqrt{a_1} + \sqrt{a_2} + \cdots + \sqrt{a_m},\ S_{\text{sqrt,even}} \coloneqq \sum_{i=1,\,2|a_i}^{m} \sqrt{a_i},\ S_{\text{sqrt,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \sqrt{a_i},$$

$$S_{\text{cbrt}} \coloneqq \sum_{i=1}^{m} \sqrt[3]{a_i} = \sqrt[3]{a_1} + \sqrt[3]{a_2} + \cdots + \sqrt[3]{a_m},\ S_{\text{cbrt,even}} \coloneqq \sum_{i=1,\,2|a_i}^{m} \sqrt[3]{a_i},\ S_{\text{cbrt,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \sqrt[3]{a_i},$$

$$S_{\text{rt},n} \coloneqq \sum_{i=1}^{m} \sqrt[n]{a_i} = \sqrt[n]{a_1} + \sqrt[n]{a_2} + \cdots + \sqrt[n]{a_m},\ S_{\text{rt,even},n} \coloneqq \sum_{i=1,\,2|a_i}^{m} \sqrt[n]{a_i},\ S_{\text{rt,odd},n} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \sqrt[n]{a_i},\ \forall n \in \mathbb{N}^\star,$$

$$S_{\text{rcpc}} \coloneqq \sum_{i=1}^{m} \frac{1}{a_i} = \frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_m},\ S_{\text{rcpc,even}} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{a_i},\ S_{\text{rcpc,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{a_i},$$

$$S_{\text{rcpc,sqr}} \coloneqq \sum_{i=1}^{m} \frac{1}{a_i^2} = \frac{1}{a_1^2} + \frac{1}{a_2^2} + \cdots + \frac{1}{a_m^2},\ S_{\text{rcpc,sqr,even}} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{a_i^2},\ S_{\text{rcpc,sqr,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{a_i^2},$$

$$S_{\text{rcpc,cb}} \coloneqq \sum_{i=1}^{m} \frac{1}{a_i^3} = \frac{1}{a_1^3} + \frac{1}{a_2^3} + \cdots + \frac{1}{a_m^3},\ S_{\text{rcpc,cb,even}} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{a_i^3},\ S_{\text{rcpc,cb,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{a_i^3},$$

$$S_{\text{rcpc,pwr},n} \coloneqq \sum_{i=1}^{m} \frac{1}{a_i^n} = \frac{1}{a_1^n} + \frac{1}{a_2^n} + \cdots + \frac{1}{a_m^n},\ S_{\text{rcpc,even,pwr},n} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{a_i^n},\ S_{\text{rcpc,odd,pwr},n} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{a_i^n},\ \forall n \in \mathbb{N}^\star,$$

$$S_{\text{rcpc,sqrt}} \coloneqq \sum_{i=1}^{m} \frac{1}{\sqrt{a_i}} = \frac{1}{\sqrt{a_1}} + \frac{1}{\sqrt{a_2}} + \cdots + \frac{1}{\sqrt{a_m}},\ S_{\text{rcpc,sqrt,even}} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{\sqrt{a_i}},\ S_{\text{rcpc,sqrt,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{\sqrt{a_i}},$$

$$S_{\text{rcpc,cbrt}} \coloneqq \sum_{i=1}^{m} \frac{1}{\sqrt[3]{a_i}} = \frac{1}{\sqrt[3]{a_1}} + \frac{1}{\sqrt[3]{a_2}} + \cdots + \frac{1}{\sqrt[3]{a_m}},\ S_{\text{rcpc,cbrt,even}} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{\sqrt[3]{a_i}},\ S_{\text{rcpc,cbrt,odd}} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{\sqrt[3]{a_i}},$$

$$S_{\text{rcpc,rt},n} \coloneqq \sum_{i=1}^{m} \frac{1}{\sqrt[n]{a_i}} = \frac{1}{\sqrt[n]{a_1}} + \frac{1}{\sqrt[n]{a_2}} + \cdots + \frac{1}{\sqrt[n]{a_m}},\ S_{\text{rcpc,even,rt},n} \coloneqq \sum_{i=1,\,2|a_i,\,a_i\neq 0}^{m} \frac{1}{\sqrt[n]{a_i}},\ S_{\text{rcpc,odd,rt},n} \coloneqq \sum_{i=1,\,2\nmid a_i}^{m} \frac{1}{\sqrt[n]{a_i}},\ \forall n \in \mathbb{N}^\star.$$

Dựa vào các công thức này, sử dụng vòng lặp `for` hoặc `while` để tính các tổng này. □

Nhận xét 1. *Nếu chỉ tính tổng S_{odd} các số lẻ của dãy $(a_i)_{i=1}^n \subset \mathbb{Z}$ thì ta có bài toán* [Vie21, 3., p. 25, Tây Ninh 2019].

Nhận xét 2 (Mở rộng $\mathbb{Z}$ ra $\mathbb{R}, \mathbb{C}$)**.** *Các tổng $S, S_{\text{sqr}}, S_{\text{cb}}, S_{\text{pwr},n}, S_{\text{sqrt}}, S_{\text{cbrt}}, S_{\text{rt},n}, S_{\text{rcpc}}$ (i.e., các tổng không có liên quan đến tính chẵn lẻ) vẫn có thể áp dụng cho các dãy số thực thay vì chỉ cho dãy số nguyên, i.e., áp dụng cho $(a_i)_{i=1}^n \subset \mathbb{R}$, $a_i \in \mathbb{R}$, $\forall i = 1, 2, \ldots, n$, thay vì chỉ cho $(a_i)_{i=1}^n \subset \mathbb{R}$, $a_i \in \mathbb{Z}$, $\forall i = 1, 2, \ldots, n$, thậm chí có thể áp dụng cho các dãy số phức $(a_i)_{i=1}^n \subset \mathbb{C}$, $a_i \in \mathbb{C}$, $\forall i = 1, 2, \ldots, n$.*

Nhận xét 3 (Mở rộng từ dãy hữu hạn dãy vô hạn & chuỗi)**.** *Bài toán trên có thể mở rộng từ dãy hữu hạn (finite sequence) ra dãy vô hạn (infinite sequence) các số nguyên $(a_n)_{i=1}^\infty \subset \mathbb{Z}$, dãy vô hạn các số thực $(a_n)_{i=1}^\infty \subset \mathbb{R}$, & dãy vô hạn các số phức $(a_n)_{i=1}^\infty \subset \mathbb{C}$, cũng như các chuỗi (series) "xác định" (i.e., có giới hạn để có thể tính được) $S \coloneqq \sum_{i=1}^\infty a_i \in \overline{\mathbb{R}}$. Dương nhiên, 1 chương trình máy tính chỉ có thể lặp (e.g., `for, while` loops) hữu hạn lần chứ không thể lặp vô hạn lần (infinite loop error) nên ta chỉ có thể tính tổng riêng S_m của 1 chuỗi S xác định để xấp xỉ chuỗi S tới 1 độ chính xác (tolerance) nào đó (tolerance thường có dạng 10^{-N} với $N \in \mathbb{N}^\star$ thích hợp), e.g.,*

$$S_m \coloneqq \sum_{i=1}^{m} a_i \to S \coloneqq \sum_{i=1}^{\infty} a_i \text{ as } n \to \infty, \text{ i.e. } \lim_{m\to\infty} S_m = S \text{ if } S \in \overline{\mathbb{R}},$$

trong đó $\overline{\mathbb{R}} \coloneqq \mathbb{R} \cup \{\pm\infty\}$ ký hiệu tập số thực mở rộng bao gồm tập số thực, âm- & dương vô cực.

Bài toán 21 ([Vie21], 5., p. 26, Nghệ An 2019, Giả thuyết Goldbach cho số nguyên tố – Goldbach conjecture for primes)**.** *Minh đố An: Cho 1 số chẵn $k \in \mathbb{N}$, $2 \le k \le 1000$, tìm 2 số nguyên tố sao cho tổng của chúng bằng số chẵn k đã cho.*

- Yêu cầu: *Viết chương trình* Pascal, Python, C/C++ *giúp An trả lời câu hỏi của Minh.*
- Dữ liệu vào: *Tệp văn bản* `prime.inp`*: Dòng đầu tiên chứa $n \in \mathbb{N}^\star$ tương ứng số test. n dòng tiếp theo, mỗi dòng chứa 1 số k, i.e., k_i, $i = 1, 2, \ldots, n$.*
- Dữ liệu ra: *Tệp văn bản* `prime.out` *gồm n dòng tương ứng n kết quả. Mỗi kết quả hiển thị tổng 2 số nguyên tố bằng số k nhập vào. E.g.:*

`prime.inp`	`prime.out`
2	$8 = 5 + 3$
8	$24 = 19 + 5$
24	

Bài toán 22 ([Vie21], 6., p. 27, Tây Ninh 2019, Số hoàn hảo – Perfect number)**.** *Số hoàn hảo là 1 số tự nhiên mà tổng tất cả các ước tự nhiên thực sự của nó bằng chính nó. Trong đó ước thực sự của 1 số là các ước dương không bằng số đó. Lập trình nhập vào 1 số tự nhiên có* 2 *chữ số bất kỳ. In ra màn hình thông báo số vừa nhập có phải là số hoàn hảo hay không? Nếu là số hoàn hảo thì in tất cả các ước của số đó.*

Bài toán 23 ([Vie21], 7., p. 27, Đồng Nai 2020, Số may mắn – Lucky number)**.** *Để động viên thành tích học tập xuất sắc của các em học sinh lớp 6-3 trong năm học 2019–2020, thầy giáo chủ nhiệm đã chuẩn bị các món quà được đánh số từ* 1 *đến* n*. Sau đó thầy giáo sẽ cho các em lên bốc thăm để nhận món quà may mắn của mình. Đầu tiên thầy giáo sẽ ghi tất cả số nguyên lẻ từ* 1 *đến* n*, sau đó sẽ ghi tất cả các số nguyên chẵn từ* 2 *đến* n *(theo thứ tự tăng dần) để tạo thành 1 dãy số phần thưởng. Mỗi bạn sẽ bốc thăm 1 số* k *ứng với con số của món quà mình đạt được.*

- Yêu cầu: *In số của món quà học sinh đạt được.*
- Dữ liệu vào: *Dòng duy nhất ghi số nguyên* n *&* k*,* $1 \leq k \leq n \leq 1000$*.*
- Dữ liệu ra: *In số của món quà học sinh đạt được:*

`lucky_number.inp`	`lucky_number.out`
10 6	2

Bài toán 24 ([Vie21], 8., p. 27, Ninh Bình 2019, Ước chung lớn nhất ƯCLN – greatest common divisor gcd)**.** *Nhập vào* 3 *số từ bàn phím, kiểm soát dữ liệu nhập vào là số nguyên dương. Lập trình tìm* ƯCLN *của* 3 *số này. E.g., nhập vào* 3 *số:* $4, 6, 12$ *thì kết quả* ƯCLN *là* 2*.*

Bài toán 25 (Bội chung nhỏ nhất BCNN – least common multiplier lcd)**.** *Nhập vào* $n \in \mathbb{N}^\star$ *số từ bàn phím, kiểm soát dữ liệu nhập vào là số nguyên dương. Lập trình tìm* ƯCLN *&* BCNN *của* n *số này.*

5 Character & String – Xâu & Chuỗi

Bài toán 26 ([Vie21], 1., p. 28, Tây Ninh 2019, Số đảo ngược – Reversed number)**.** *Tìm số đảo ngược* y *của 1 số* $x \in \mathbb{Z}$ *biết* y *gồm các chữ số của* x *& viết theo thứ tự ngược lại. Xuất ra kết quả là số* $y \mod 19$*.* Dữ liệu: $x \in \mathbb{N}^\star$*.* Kết quả: $y \mod 19$ *với* y *là số đảo ngược của* x*.*

`reversed_number.inp`	`reversed_number.out`	Giải thích
123	17	Đảo ngược của 123 là 321 & $321 \mod 19 = 17$

Bài toán 27 ([Vie21], 2., pp. 28–29, Bắc Giang 2020, Nén xâu – String compression)**.** *Viết chương trình nhập vào 1 xâu ký tự chỉ gồm các chữ cái Tiếng Anh, chữ số, dấu cách, & dấu gạch nối.*

- Yêu cầu: *Nén các ký tự liên tiếp giống nhau thành số lượng ký tự & ký tự đó, rồi đưa ra xâu sau khi nén.*
- Dữ liệu vào: *Đọc từ file văn bản* `string_compression.inp` *gồm 1 xâu* S *có số lượng ký tự không quá* 255 *ký tự.*
- Dữ liệu ra: *Đưa ra file văn bản* `string_compression.out` *gồm xâu* S *sau khi nén.*

`string_compression.inp`	`lstring_compression.out`
aaaababb cc	4a1b1a2b4 2c

Bài toán 28 ([Vie21], 3., p. 30, Hậu Giang 2020, Tính nhân – Multiplication)**.** *Viết chương trình nhập vào 2 số nguyên dương* $a, b \in \mathbb{N}^\star$*. Sau đó thực hiện nhân* $a \times b$ *như cách nhân bằng tay thông thường ở tiểu học. E.g., nhập vào thừa số thứ 1:* 125*, nhập vào thừa số thứ 2:* 15*.* Dữ liệu ra:

```
  125
x
   15
-----
  625
 125
-----
 1875
```

6 1D Array & List – Mảng 1 Chiều & Danh Sách

7 Matrix – Ma Trận

8 Arrangement – Sắp Xếp

See, e.g., [Knu98, Chap. 5: Sorting], [Vie21, Chap. II, Sect. Dạng bài sắp xếp].

Bài toán 29 ([Vie21], 1., p. 83, Bắc Giang 2019, Dãy số – Sequence)**.** *Sử dụng hàm Randomize để khởi tạo dãy số ngẫu nhiên từ* 0 *đến* 9 *gồm* $n \in \mathbb{N}^\star$ *phần tử,* $0 < n \leq 100$*, kết quả ghi ra tệp* `random.out`*, mỗi phần tử cách nhau* 1 *dấu cách.*

- Yêu cầu: *Viết chương trình đọc dữ liệu từ tệp* `random.inp`*, sau đó sắp xếp lại các phần tử theo chiều tăng dần, đồng thời cho biết số lần xuất hiện của mỗi phần tử trong dãy số đã được khởi tạo.*
- Dữ liệu ra: *Ghi ra tệp* `random.out` *gồm* 11 *dòng: Dòng thứ nhất là dãy các phần tử đã được sắp xếp. Dòng thứ* 2 *đến dòng thứ* 11 *tương ứng chữ số ghi tổng số lần xuất hiện của* $0, 1, \ldots, 9$*. E.g.:*

`random_sequence.inp`	`random_sequence.out`
0 5 2 0 1 6 7 8 7 3 1	0 0 1 1 2 3 5 6 7 7 8 2 2 1 1 0 1 1 2 1 0

Bài toán 30 ([Vie21], 2., p. 85, Vị Thanh, Hậu Giang 2019, Dãy số không giảm – Nondecreasing sequence)**.** *Nhập từ bàn phím* 3 *số nguyên dương* a_1, a_2, a_3*,* $100 < a_1, a_2, a_3 < 10^5$*. Dãy số b được sinh ra bằng cách ghép từng số nguyên dương đã nhập lần lượt với 2 số còn lại, e.g.,* $a_1 = 234$*,* $a_2 = 123$*,* $a_3 = 345$ *ta tìm được* $b_1 = 234123$*,* $b_2 = 234345$*,* $b_3 = 123234$*,* $b_4 = 123345$*,* $b_5 = 345234$*. Sắp xếp các số trong dãy số b thành dãy không giảm & xuất ra màn hình, các số cách nhau 1 khoảng trắng. E.g.:*

`nondecreasing_sequence.inp`	`nondecreasing_sequence.out`
$a_1 = 234$ $a_2 = 123$ $a_3 = 345$	123234 123345 234123 234345 345123 345234

Bài toán 31 ([Vie21], 1., p. 87, Sorting ascending)**.** *Cho vào* m *dãy số nguyên* $(a_{i,j})_{j=1}^{n_i}$*,* $n_i \in \mathbb{N}^\star$*,* $n_i \leq 100$*,* $\forall i = 1, 2, \ldots, m$*,* $|a_{ij}| < 32000$*,* $\forall i = 1, 2, \ldots, m$*,* $\forall j = 1, 2, \ldots, \max\{n_i | i = 1, 2, \ldots, m\}$*.*

- Yêu cầu: *Sắp xếp từng dãy số trên theo thứ tự tăng dần.*
- Dữ liệu vào: *Trong* m *dòng, mỗi dòng là dãy số, bắt đầu là 1 số nguyên* n *là số lượng các phần tử của dãy số,* $1 \leq n \leq 100$*,* n *số nguyên tiếp theo là giá trị các phần tử của dãy.*
- Dữ liệu ra: *Ghi ra* m *dòng là* m *dãy số đã được sắp xếp theo thứ tự tăng dần. E.g.:*

`sorting_ascending.inp`	`sorting_ascending.out`
2 2 1 3 4 3 1 4 1 4 5 2	1 2 1 3 4 1 2 4 5

9 Algorithm – Thuật Toán

9.1 Recursion algorithm – Thuật toán đệ quy

Đệ quy (recursion) là phương pháp dùng trong các chương trình máy tính trong đó có 1 hàm tự gọi chính nó.

Bài toán 32 ([Vie21], p. 91, Giai thừa – Factorial $n!$)**.** *Tính* $n! = \prod_{i=1}^{n} i = 1 \cdot 2 \cdots (n-1)n$*.*

- Input: *Dòng đầu là số lượng test. Mỗi dòng tiếp theo gồm 1 số* $n \in \mathbb{N}$*.*

- Output: *Với mỗi test, in ra $n!$ theo mẫu:*

factorial.inp	factorial.out
2	$3! = 6$
3	$4! = 24$
4	

Bài toán 33 ([Vie21], 1., p. 92, Hàm $f91$ của McCarthy – McCarthy's $f91$ function)**.** *McCarthy là 1 nhà khoa học máy tính nổi tiếng, ông đã định nghĩa hàm đệ quy $f91 : \mathbb{Z} \to \mathbb{Z}$ như sau:*

$$f_{91}(n) = \begin{cases} f_{91}(f_{91}(n+11)), & \text{if } n \leq 100, \\ n-1, & \text{if } n \geq 101. \end{cases}$$

Viết 1 chương trình tính toán hàm McCarthy's f91.

- Input: *File input chứa 1 dãy các số nguyên dương, mỗi số không quá* 1000*. Mỗi số trên 1 dòng.*
- Output: *Hiện ra theo định dạng trong ví dụ sau:*

McCarthy_f91_function.inp	McCarthy_f91_function.out
500	f91(500) = 490
91	f91(91) = 91

See, e.g., Wikipedia/McCarthy 91 function.

Bài toán 34 ([Vie21], 2., p. 93, Chỉnh hợp – Arrangement A_n^k)**.** *Tìm tất cả các chỉnh hợp chập k của n phần tử từ* 1 *đến n, $0 < k \leq n < 10$.*

- Input: *File input chứa 1 dòng gồm $n, k \in \mathbb{N}$.*
- Output: *In ra số chỉnh hợp tìm được & liệt kê các chỉnh hợp, giữa các test là 1 dòng trắng, e.g.,*

arrangement.inp	arrangement.out	arrangement.inp	arrangement.out
1 3	3 11 13 33	2 3	6 1 2 1 3 2 1 2 3 3 1 3 2

Bài toán 35 ([Vie21], 3., p. 93, Hoán vị – Permutation $P_n = n!$)**.** *Viết chương trình in ra tất cả các hoán vị của $n \in \mathbb{N}^\star$ được nhập từ bàn phím, e.g.,*

permutation.inp	permutation.out
2	1 2 2 1
3	1 2 3 1 3 2 2 1 3 2 3 1 3 2 1 3 1 2

Bài toán 36 ([Vie21], 5., p. 94, Tổ hợp – Combinatoric C_n^k)**.** *Tìm tất cả các tổ hợp chập k của n phần tử từ* 1 *đến n, $0 < k \leq n < 10$.*

- Input: *File input có dòng đầu gồm 1 số tự nhiên* `ntest` *là số lượng test của file* `combinatoric.inp`*. Mỗi test 1 dòng gồm 2 số $n, k \in \mathbb{Z}$.*
- Output: *Với mỗi test, in ra số tổ hợp tìm được & liệt kê các tổ hợp, giữa các test là 1 dòng trắng. E.g.:*

combinatoric.inp	combinatoric.out
2 1 3 2 3	3 1 2 3 3 1 2 1 3 2 3

Bài toán 37 ([Vie21], 4., pp. 93–94, Trò chơi số học – Number theory game)**.** *1 trò chơi phổ biến của trẻ em với bảng $n \times n$ ô $2 \leq n \leq 5$. Trong mỗi ô chứa 1 số có 1 chữ số từ 1 tới 9. Với 1 số số cho trước, điền các số còn lại vào ô sao cho tổng các hàng ngang bằng nhau & bằng tổng các hàng dọc.*

- Input: `number_theory_game.inp`*: Số đầu tiên là 1 số nguyên* `ntest` *là số lượng test của bài, mỗi test gồm có: Dòng đầu tiên là số n. Tiếp theo là ma trận $n \times n$ trong đó số 0 là ô trắng cần điền.*
- Output: `number_theory_game.out`*: Với mỗi test, nếu có thể tìm ra đáp án thỏa mãn quy luật của bảng thì in ra ma trận $n \times n$ 1 cách điền bất kỳ. Nếu không có kết quả nào in ra 1 chuỗi:* `"cannot find."`*. Giữa các test cách nhau 1 dòng trắng. E.g.:*

number_theory_game.inp	number_theory_game.out
1 3 1 0 5 0 3 0 3 0 4	1 4 5 6 3 1 3 3 4

9.2 Search algorithm – Thuật toán tìm kiếm

Tìm kiếm & sắp xếp là 2 hoạt động hằng ngày ta thường sử dụng trong các ứng dụng tin học.
See, e.g., [Knu98, Chap. 6: Searching].

10 Problem in Elementary Physics – Bài Toán Tin Học Trong Vật Lý Sơ Cấp

11 Problem in Elementary Chemistry – Bài Toán Tin Học Trong Hóa Học Sơ Cấp

12 Miscellaneous

Bài toán 38 ([BTC10], 1., p. 5, Connect)**.** *Cho n số nguyên dương $a_1, a_2, \ldots, a_n$, $n \in \mathbb{N}$, $1 < n \leq 100$, $0 < a_i \leq 10^9$, $\forall i = 1, 2, \ldots, n$. Từ các số nguyên này người ta tạo ra 1 số nguyên mới bằng cách kết nối tất cả các số đã cho viết liên tiếp nhau. E.g., với $n = 4$ & các số $12, 34, 567, 890$ ta có thể tạo ra các số mới như sau:* 1234567890*,* 3456789012*,* 8905673412*, ... Dễ thấy có $4! = 24$ cách tạo mới như vậy. Trong trường hợp này, số lớn nhất có thể tạo thành là* 8905673412*.*

- Yêu cầu: *Cho n & các số $a_1, a_2, \ldots, a_n$. Xác định số lớn nhất có thể kết nối được theo quy tắc trên.*
- Dữ liệu vào: *Cho trong file văn bản* `connect.inp` *gồm $n + 1$ dòng. Dòng đầu tiên ghi số nguyên n. Trong các dòng còn lại, dòng thứ $i + 1$ ghi số a_i.*
- Dữ liệu ra: *Ghi vào file văn bản* `connect.out` *số lớn nhất được kết nối thành từ các số ban đầu. E.g.:*

connect.inp	connect.out
4 12 34 567 890	8905673412

13 Google Kickstart Round A 2020

Watch YouTube/William Lin/Winning Google Kickstart Round A 2020.

Problem 1 (Google Kickstart Round A 2020, Allocation). *There are n houses for sale. The ith house costs a_i dollars to buy. You have a budget of b dollars to spend. What is the maximum number of houses you can buy?*

- Input: *The 1st line of the input gives the number of test cases, t. t test cases follow. Each test case begins with a single line containing the 2 integers n, b. The 2nd line contains n integers. The ith integer is a_i, the cost of the ith house.*
- Output: *For each test case, output 1 line containing* `Case #x: y`, *where* **`x`** *is the test case number (starting from 1) &* **`y`** *is the maximum number of houses you can buy.*
- Limits: *Time limit:* 15 s/test set. *Memory limit:* 1GB. $1 \le t \le 100$, $1 \le b \le 10^5$, $1 \le a_i \le 1000$, $\forall i = 1, 2, \ldots, n$. *Test set 1:* $1 \le n \le 100$. *Test set 2:* $1 \le n \le 10^5$.
- Sample:

`allocation.inp`	`allocation.out`
3 4 100 20 90 40 90 4 50 30 30 10 10 3 300 999 999 999	Case #1: 2 Case #2: 3 Case #3: 0

Problem 2 (Google Kickstart Round A 2020, Plates). *Dr. Patel has n stacks of plates. Each stack contains k plates. Each plate has a positive beauty value, describing how beautiful it looks. Dr. Patel would like to take exactly p plates to use for dinner tonight. If he would like to take a plate in a stack, he must also take all of the plates above it in that stack as well. Help Dr. Patel pick the p plates that would maximize the total sum of beauty values.*

- Input: *The 1st line of the input gives the number of test cases, t. t test cases follow. Each test case begins with a line containing the 3 integers n, k, p. Then, n lines follow. The ith line contains k integers, describing the beauty values of each stack of plates from top to bottom.*
- Output: *For each test case, output 1 line containing* `Case #x: y`, *where* **`x`** *is the test case number (starting from 1) &* **`y`** *is the maximum total sum of beauty values that Dr. Patel could pick.*
- Limits: *Time limit:* 20 s/test set. *Memory limit:* 1GB. $\le t \le 100$, $1 \le k \le 30$, $1 \le p \le nk$. *The beauty values are between 1 & 100, inclusive. Test set 1:* $1 \le n \le 3$. *Test set 2:* $1 \le n \le 50$.
- Sample:

`plate.inp`	`plate.out`
2 2 4 5 10 10 100 30 80 50 10 50 3 2 3 80 80 15 50 20 10	Case #1: 250 Case #2: 180

14 Resources

[Đàm+09a; Đàm+09b; Đàm+11; Knu97; Vie21; Vie22].

Tài liệu

[BTC10] BTC. *Tuyển Tập Đề Thi Olympic 30 Tháng 4, Lần Thứ XVI - 2010 Tin học*. Nhà Xuất Bản Đại Học Sư Phạm, 2010, p. 285.

[Đàm+09a] Hồ Sĩ Đàm, Đỗ Đức Đông, Lê Minh Hoàng, and Nguyễn Thanh Hùng. *Tài Liệu Giáo Khoa Chuyên Tin, quyển 1*. Nhà Xuất Bản Giáo Dục Việt Nam, 2009, p. 219.

[Đàm+09b] Hồ Sĩ Đàm, Đỗ Đức Đông, Lê Minh Hoàng, and Nguyễn Thanh Hùng. *Tài Liệu Giáo Khoa Chuyên Tin, quyển 2*. Nhà Xuất Bản Giáo Dục Việt Nam, 2009, p. 240.

[Đàm+11] Hồ Sĩ Đàm, Đỗ Đức Đông, Lê Minh Hoàng, and Nguyễn Thanh Hùng. *Tài Liệu Giáo Khoa Chuyên Tin, quyển 3*. Nhà Xuất Bản Giáo Dục Việt Nam, 2011, p. 170.

[Knu97] Donald Ervin Knuth. *The Art of Computer Programming. Volume 1: Fundamental Algorithms*. 3rd edition. Addison-Wesley Professional, 1997, pp. xx+652.

[Knu98] Donald Ervin Knuth. *The Art of Computer Programming. Volume 3: Sorting and Searching*. 2nd edition. Addison-Wesley Professional, 1998, pp. xiii+782.

[Vie21] Học Viện VietSTEM. *Sách Luyện Thi Hội Thi Tin Học Trẻ với Python Bảng B: Thi Kỹ Năng Lập Trình Cấp Trung Học Cơ Sở*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2021, p. 190.

[Vie22] Học Viện VietSTEM. *Lập Trình với Python: Hành Trang Cho Tương Lai*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2022, p. 224.