

Lecture Note: Information Technology Fundamentals

Bài Giảng: Nền Tảng Công Nghệ Thông Tin

Nguyễn Quân Bá Hồng*

Ngày 15 tháng 5 năm 2025

Tóm tắt nội dung

This text is a part of the series *Some Topics in Advanced STEM & Beyond*:

URL: https://nqbh.github.io/advanced_STEM/.

Latest version:

- *Lecture Note: Information Technology Fundamentals – Bài Giảng: Nền Tảng Công Nghệ Thông Tin.*

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/lecture/NQBH_IT_fundamentals_lecture.pdf.

TeX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/lecture/NQBH_IT_fundamentals_lecture.tex.

- *Codes.*

PDF: URL: [.pdf](#).

TeX: URL: [.tex](#).

Mục lục

1 Basic	1
1.1 Basic guides	1
2 Miscellaneous	3
Tài liệu	3

1 Basic

Resources – Tài nguyên.

1. [Hạ22]. BÙI VIỆT HÀ. *Python Cơ Bản*.
2. [Hạ21]. BÙI VIỆT HÀ. *Lời Giải Bài Tập Python Cơ Bản*.
3. [Hạ23]. BÙI VIỆT HÀ. *Python Nâng Cao*.
4. [Mat19; Mat23]. ERIC MATTHES. *Python Crash Course: A Hands-on, Project-Based Introduction to Programming*.

1.1 Basic guides

- Truy cập trang web Google Colab: <https://colab.research.google.com>.
- *Cách tạo 1 file Jupyter notebook mới*: File → New notebook in Drive.
- Thêm ghi chú: Nhấp vào + Text → Gõ ghi chú, comment → Shift Enter.
- Thêm code: Nhấp vào + Code → Gõ code vào → Run.
- `int`: integer: kiểu dữ liệu số nguyên $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$.
- `float`: real number: integer: kiểu dữ liệu số thực \mathbb{R} .

Bài toán 1. *Viết chương trình Python để tính tổng, hiệu, tích, thương của 2 số $a, b \in \mathbb{R}$ được nhập từ bàn phím.*

*A scientist- & creative artist wannabe, a mathematics & computer science lecturer of Department of Artificial Intelligence & Data Science (AIDS), School of Technology (SOT), UMT Trường Đại học Quản lý & Công nghệ TP.HCM, Hồ Chí Minh City, Việt Nam.
E-mail: nguyenquanbahong@gmail.com & hong.nguyenquanba@umt.edu.vn. Website: <https://nqbh.github.io/>. GitHub: <https://github.com/NQBH>.

Chứng minh. Python:

```
a = float(input("a = "))
b = float(input("b = "))
print("Sum a + b = ", a + b)
print("Difference a - b = ", a - b)
print("Product ab = ", a * b)
if b == 0:
    print("Division by zero error")
else:
    print("Quotient a/b = ", a / b)
```

□

Remark 1. Nếu không xét trường hợp $b = 0$ thì sẽ bị lỗi chia cho 0:

Traceback (most recent call last):

```
File "/home/nqhb/advanced_STEM_beyond/IT_fundamentals/Python/basic.py", line 6, in <module>
    print("Quotient a/b = ", a / b)
    ~~~~
```

ZeroDivisionError: float division by zero

Bài toán 2 (Even & odd – Chẵn & lẻ, +0.5). Viết chương trình Python để xét tính chẵn lẻ của 1 số $a \in \mathbb{Z}$ được nhập từ bàn phím.

Bài toán 3 (Divisible by – Tính chia hết, +0.5). Viết chương trình Python để xét xem $a \in \mathbb{Z}$ có chia hết cho $b \in \mathbb{Z}$ không, với a, b được nhập từ bàn phím. Nếu có thông báo `a is divisible by b`, nếu không thì in số dư r của phép chia a cho b , với $0 \leq r < |b|$.

Bài toán 4. Cho $n \in \mathbb{N}^*$ được nhập từ bàn phím. Viết chương trình Python để tính: (a) Tổng của n số nguyên dương đầu tiên: $\sum_{i=1}^n i = 1 + 2 + \dots + n$ & so sánh với $\frac{n(n+1)}{2}$. (b) Tổng của n số nguyên dương lẻ đầu tiên: $\sum_{i=1}^n (2i-1) = 1 + 3 + 5 + \dots + (2n-1)$ & so sánh với n^2 . (c) Tổng của n số lẻ nguyên dương chẵn đầu tiên: $\sum_{i=1}^n 2i = 2 + 4 + 6 + \dots + 2n$ & so sánh với $n(n+1)$. (d) Tổng bình phương của n số nguyên dương đầu tiên: $\sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2$ & so sánh với $\frac{n(n+1)(2n+1)}{6}$. (e) Tổng bình phương của n số nguyên dương lẻ đầu tiên: $\sum_{i=1}^n (2i-1)^2 = 1^2 + 3^2 + 5^2 + \dots + (2n-1)^2$. (f) Tổng bình phương của n số lẻ nguyên dương chẵn đầu tiên: $\sum_{i=1}^n (2i)^2 = 2^2 + 4^2 + 6^2 + \dots + (2n)^2$ & so sánh với $\frac{2n(n+1)(2n+1)}{3}$. (g) Tổng lập phương của n số nguyên dương đầu tiên: $\sum_{i=1}^n i^3 = 1^3 + 2^3 + \dots + n^3$ & so sánh với $\frac{n^2(n+1)^2}{4}$. (e) Tổng lập phương của n số nguyên dương lẻ đầu tiên: $\sum_{i=1}^n (2i-1)^3 = 1^3 + 3^3 + 5^3 + \dots + (2n-1)^3$. (f) Tổng lập phương của n số nguyên dương chẵn đầu tiên: $\sum_{i=1}^n (2i)^3 = 2^3 + 4^3 + 6^3 + \dots + (2n)^3$ & so sánh với $2n^2(n+1)^2$.

Bài toán 5 (Triangle – Tam giác). Viết chương trình Python để xét 3 số $a, b, c \in (0, \infty)$ được nhập từ bàn phím có phải là: (a) 3 cạnh của 1 tam giác hay không nhờ bất đẳng thức 3 cạnh tam giác $a < b + c, b < c + a, c < a + b$. (b) Nếu a, b, c là 3 cạnh tam giác, phân loại tam giác đó: tam giác nhọn, tam giác vuông, tam giác tù, tam giác cân, tam giác đều, tam giác vuông cân.

Bài toán 6. Viết chương trình tính chu vi, diện tích, 3 đường cao của 1 tam giác với độ dài 3 cạnh $a, b, c \in (0, \infty)$ được nhập từ bàn phím.

Bài toán 7. Cho 1 mảng số thực a_1, \dots, a_n được nhập từ bàn phím ứng với số tiền thu được mỗi tháng, trong đó $a_i < 0$: lỗ, $a_i = 0$: huê vốn, $a_i > 0$: lời/lãi. Viết chương trình Python xuất ra màn hình: (a) Số tháng lời, lỗ, huê vốn. (b) Tổng số tiền lời, tổng số tiền lỗ, tổng số thu nhập cuối cùng (sau khi lấy tổng số tiền lời – tổng số tiền lỗ).

Input. Dãy số thực a_1, \dots, a_n .

Output. Số tháng lời, số tháng lỗ, số tháng huê vốn. Dòng tiếp theo: Tổng số tiền lời, tổng số tiền lỗ.

Sample.

money.inp	money.out
10.5 -2.3 3.6 4.5 -7.41 0 1.23	4 2 1
	19.83 -9.71
	10.12

Bài toán 8 ([Thá+25], VD3, p. 25, tính tiền vốn lẫn lãi nếu không rút tiền ra). 1 người gửi số tiền $m_0 \in (0, \infty)$ đồng vào ngân hàng với lãi suất $r\%/năm$. Biết nếu không rút tiền ra khỏi ngân hàng thì cứ sau mỗi năm, số tiền lãi sẽ được nhập vào vốn ban đầu. Biết số tiền nhận được (bao gồm cả vốn lẫn lãi) sau n năm là

$$M(m_0, r, n) = m_0 \left(1 + \frac{r}{100}\right)^n \text{ đồng},$$

nếu trong khoảng thời gian này người gửi không rút tiền ra & lãi suất không thay đổi. Viết chương trình Python để: (a) Tính $M(m_0, r, n)$ với m_0, r, n lần lượt được nhập vào. (b) Xuất ra số tiền nhận được (bao gồm cả vốn lẫn lãi) sau năm 1, năm 2, ..., năm n , i.e., xuất ra dãy số thực $\{M(m_0, r, i)\}_{i=1}^n = M(m_0, r, 1), M(m_0, r, 2), \dots, M(m_0, r, n)$. (c) Với số tiền m được nhập từ bàn phím, cho biết sau bao nhiêu năm thì số tiền cả vốn lẫn lãi vượt qua số tiền m kỳ vọng này.

Input. Dòng 1 chứa lần lượt $m_0, r \in (0, \infty)$, $n \in \mathbb{N}^*$.

Output. Dòng 1 chứa $T(m_0, r, n)$. Dòng 2 chứa dãy số $\{M(m_0, r, i)\}_{i=1}^n = M(m_0, r, 1), M(m_0, r, 2), \dots, M(m_0, r, n)$.

Sample.

rate.inp	rate.out
125000000 4.9 3	144290081.125

Bài toán 9 ([Thá+25], 10., p. 30, tính tiền vốn lẫn lãi nếu không rút tiền ra). Giả sử năm đầu tiên, A gửi vào ngân hàng $m_0 \in (0, \infty)$ đồng với lãi suất $r\%$ năm. Hết năm đầu tiên, A không rút tiền ra & gửi thêm m_0 đồng nữa. Hết năm thứ 2, A cũng không rút tiền ra & lại gửi thêm m_0 đồng nữa. Cứ tiếp tục như vậy cho các năm sau. Biết số tiền cả vốn lẫn lãi mà A có được sau $n \in \mathbb{N}^*$ năm là

$$M(m_0, r, n) = \frac{m_0(100 + r)}{r} \left[\left(1 + \frac{r}{100}\right)^n - 1 \right] \text{ đồng},$$

nếu trong khoảng thời gian này lãi suất không thay đổi. Viết chương trình Python để: (a) Tính $M(m_0, r, n)$ với m_0, r, n lần lượt được nhập vào. (b) Xuất ra số tiền nhận được (bao gồm cả vốn lẫn lãi) sau năm 1, năm 2, ..., năm n , i.e., xuất ra dãy số thực $\{M(m_0, r, i)\}_{i=1}^n = M(m_0, r, 1), M(m_0, r, 2), \dots, M(m_0, r, n)$. (c) Với số tiền m được nhập từ bàn phím, cho biết sau bao nhiêu năm thì số tiền cả vốn lẫn lãi vượt qua số tiền m kỳ vọng này.

Input. Dòng 1 chứa lần lượt $m_0, r \in (0, \infty)$, $n \in \mathbb{N}^*$.

Output. Dòng 1 chứa $T(m_0, r, n)$. Dòng 2 chứa dãy số $\{M(m_0, r, i)\}_{i=1}^n = M(m_0, r, 1), M(m_0, r, 2), \dots, M(m_0, r, n)$.

Sample.

rate.inp	rate.out
125000000 4.9 5	723102450.785

Bài toán 10 ([Thá+25], 11., p. 30, tính tiền vốn lẫn lãi nếu không rút tiền ra). 1 người gửi số tiền $m_0 \in (0, \infty)$ đồng vào ngân hàng. Biểu lãi suất của ngân hàng như sau: Chia mỗi năm thành $m \in \mathbb{N}^*$ kỳ hạn & lãi suất $r\%/năm$. Biết nếu không rút tiền ra khỏi ngân hàng thì cứ sau mỗi kỳ hạn, số tiền lãi sẽ được nhập vào vốn ban đầu. Biết số tiền nhận được (bao gồm cả vốn lẫn lãi) sau n năm gửi là

$$M(m_0, r, m, n) = m_0 \left(1 + \frac{r}{100m}\right)^{mn},$$

nếu trong khoảng thời gian này người gửi không rút tiền ra & lãi suất không thay đổi. Viết chương trình Python để: (a) Tính $M(m_0, r, m, n)$ với m_0, r, m, n lần lượt được nhập vào. (b) Xuất ra số tiền nhận được (bao gồm cả vốn lẫn lãi) sau năm 1, năm 2, ..., năm n , i.e., xuất ra dãy số thực $\{M(m_0, r, m, i)\}_{i=1}^n = M(m_0, r, m, 1), M(m_0, r, m, 2), \dots, M(m_0, r, m, n)$. (c) Với số tiền M được nhập từ bàn phím, cho biết sau bao nhiêu năm thì số tiền cả vốn lẫn lãi vượt qua số tiền M kỳ vọng này.

Input. Dòng 1 chứa lần lượt $m_0, r \in (0, \infty)$, $m, n \in \mathbb{N}^*$.

Output. Dòng 1 chứa $T(m_0, r, m, n)$. Dòng 2 chứa dãy số $\{M(m_0, r, m, i)\}_{i=1}^n = M(m_0, r, m, 1), M(m_0, r, m, 2), \dots, M(m_0, r, m, n)$.

2 Miscellaneous

1. Install Anaconda for Python: <https://www.anaconda.com/>.

Tài liệu

- [Hà21] Bùi Việt Hà. *Lời Giải Bài Tập Python Cơ Bản*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2021, p. 144.
- [Hà22] Bùi Việt Hà. *Python Cơ Bản*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2022, p. 254.
- [Hà23] Bùi Việt Hà. *Python Nâng Cao*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2023, p. 354.
- [Mat19] Eric Matthes. *Python Crash Course: A Hands-on, Project-Based Introduction to Programming*. 2nd edition. No Starch Press, 2019, pp. xxxvi+506.

- [Mat23] Eric Matthes. *Python Crash Course: A Hands-on, Project-Based Introduction to Programming*. 3rd edition. No Starch Press, 2023, pp. xxxvi+506.
- [Thá+25] Đỗ Đức Thái, Phạm Xuân Chung, Nguyễn Sơn Hà, Nguyễn Thị Phương Loan, Phạm Sỹ Nam, Phạm Minh Phương, and Phạm Hoàng Quân. *Chuyên Đề Học Tập Toán 10*. Nhà Xuất Bản Đại Học Sư Phạm, 2025, p. 72.