

Lecture Note: Information Technology Fundamentals

Bài Giảng: Nền Tảng Công Nghệ Thông Tin

Nguyễn Quân Bá Hồng*

Ngày 19 tháng 6 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 Python Guides – Hướng Dẫn Lập Trình Python Cơ Bản	1
1.1 Input & output – Nhập xuất dữ liệu	2
1.2 Loop commands – Các lệnh lặp	2
2 Recursion – đệ Quy	5
3 Miscellaneous	5
Tài liệu	6

1 Basic Python Guides – Hướng Dẫn Lập Trình Python Cơ Bản

Resources – Tài nguyên.

1. [Đúc22]. NGUYỄN TIẾN ĐỨC. *Tuyển Tập 200 Bài Tập Lập Trình Bằng Ngôn Ngữ Python*.
2. [Hà22]. BÙI VIỆT HÀ. *Python Cơ Bản*.
3. [Hà21]. BÙI VIỆT HÀ. *Lời Giải Bài Tập Python Cơ Bản*.
4. [Hà23]. BÙI VIỆT HÀ. *Python Nâng Cao*.
5. [Mat19; Mat23]. ERIC MATTHES. *Python Crash Course: A Hands-on, Project-Based Introduction to Programming*.
6. [Quế20]. TRẦN THÔNG QUẾ. *Bài Tập Lập Trình Cơ Bản Với Ngôn Ngữ Python*.

*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>.

1.1 Input & output – Nhập xuất dữ liệu

- Nhập kiểu dữ liệu dạng chuỗi (string), e.g., tên, từ, câu, etc.: `a = input()`.
- Nhập kiểu dữ liệu số nguyên: `a = int(input())`.
- Nhập kiểu dữ liệu số nguyên: `a = float(input())`.
- 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.

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 $0 \leq r < |b|$.

1.2 Loop commands – Các lệnh lặp

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 $\{M(m_0, r, m, i)\}_{i=1}^n = M(m_0, r, m, 1), M(m_0, r, m, 2), \dots, M(m_0, r, m, n)$.

Bài toán 11 (Even vs. odd – chẵn lẻ). Tính số số chẵn & số số lẻ của 1 dãy gồm n số nguyên cho trước.

Input. Dòng 1 chứa $n \in \mathbb{N}^*$. Dòng 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Output. In ra số số chẵn $a_i : 2$ & số số lẻ $a_i \not\equiv 2$.

Sample.

even_odd.inp	even_odd.out
10	number of evens: 4
1 -2 3 5 0 7 -100 -42 7 6345	number of odds: 6

Python code:

Bài toán 12 (Population prediction – Dự báo dân số). Giả sử tỷ lệ tăng dân số bình quân hàng năm của Việt Nam là $r \in (0, \infty)$ & dân số hiện nay là P_0 . Tính dân số sau $n \in \mathbb{N}^*$ năm. Tính số năm ít nhất để dân số đạt $P \in (0, \infty)$ cho trước.

Input. Dòng 1 chứa dân số ban đầu $P_0 \in (0, \infty)$, tỷ lệ tăng dân số bình quân $r \in (0, \infty)$, số năm $n \in \mathbb{N}^*$. Dòng 2 chứa số dân P cần đạt.

Output. Dòng 1 chứa số dân P_n sau n năm. Dòng 2 chứa số dân tăng từng năm tới năm n, i.e., in ra dãy số thực P_0, P_1, \dots, P_n . Dòng 3 chứa số năm ít nhất để dân số đạt ngưỡng P.

Sample.

population.inp	population.out
96525000 0.003 5	97981588.35
100000000	96525000.00 96814575.00 97105018.72 97396333.78 97688522.78 97981588.35
	12

Bài toán 13 ([DT06], VD2, p. 15). (a) Tìm GTLN $\max\{a, b, c\}$ & GTNN $\min\{a, b, c\}$ với $a, b, c \in \mathbb{R}$. (b) Mở rộng bài toán cho $n \in \mathbb{N}^*$ số a_1, a_2, \dots, a_n .

Bài toán 14 ([DT06], VD3, p. 16). (a) Tính giá trị của hàm số $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ xác định bởi công thức:

$$f(x, y) = \begin{cases} x^3 + y^3 & \text{if } 0 \leq x - y \leq 10, \\ x^2 + y^2 & \text{if } x - y < 0, y \geq 0, \\ (x - y)^2 & \text{else.} \end{cases}$$

(b) Mở rộng bài toán cho các hàm số tương tự.

Bài toán 15 ([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 nguyên dương của số đó (i.e., bao gồm tất cả các ước tự nhiên thực sự & chính số đó).

Bài toán 16 ([Tru23], 1., p. 20, HSG Lớp 12 Nam Định 2020–2021, Nearly perfect number – Số gần hoàn hảo). 1 số nguyên dương $a \in \mathbb{N}^*$ được gọi là số “gần hoàn hảo” nếu thỏa mãn điều kiện: $2a \leq k$ với k là tổng các ước số của a , e.g., 12 là 1 số “gần hoàn hảo” vì $2 \cdot 12 < 1 + 2 + 3 + 4 + 6 + 12$ ($24 < 28$).

- Input. Dòng đầu tiên chứa số nguyên dương $n \in \mathbb{N}^*$, $1 \leq n \leq 10^4$. n dòng tiếp theo, mỗi dòng là 1 số nguyên dương có giá trị $\leq 10^6$.
- Output. Dòng đầu tiên ghi số lượng số “gần hoàn hảo”. Các dòng tiếp theo, mỗi dòng ghi 1 số “gần hoàn hảo”, số gặp trước thì viết trước.
- Sample.

near_perfect_number.inp	near_perfect_number.out
5	2
8	12
16	6
12	
6	
7	

2 Recursion – Đệ Quy

Bài toán 17. Viết hàm đệ quy tính $n!$ với $n \in \mathbb{N}$ được nhập vào từ bàn phím.

Bài toán 18. Với $n \in \mathbb{N}$ được nhập vào, viết hàm đệ quy để: (a) In ra dãy số Fibonacci $\{F_i\}_{i=0}^n$. (b) In ra số Fibonacci F_n .

Bài toán 19 (Ha Noi tower problem – Bài toán tháp Hà Nội). Thực hiện chuyển n đĩa từ cọc 1 sang cọc 2, được dùng cọc trung gian. Luật chơi: (i) Mỗi lần chỉ chuyển 1 đĩa. (ii) Đĩa nhỏ nằm trên đĩa lớn. (iii) Trong lúc chuyển không được để đĩa lên bất kỳ chỗ nào.

Bài toán 20 ([Quế20], 2., p. 89). Nhập $n \in \mathbb{N}, n \geq 2$ số vào từ bàn phím. Viết chương trình Python sử dụng đệ quy để tìm ƯCLN, BCNN của n số đó.

Bài toán 21 ([Quế20], 3., p. 89). Cho dãy $\{a_n\}_{n=0}^\infty$ được định nghĩa bởi công thức truy hồi:

$$a_0 = 1, a_n = n^2 a_0 + (n-1)^2 a_1 + (n-2)^2 a_2 + \cdots + 1^2 a_{n-1}, \forall n \in \mathbb{N}^*.$$

Viết chương trình Python sử dụng đệ quy để tính a_n với $n \in \mathbb{N}$ được nhập vào từ bàn phím.

Bài toán 22 (Mở rộng [Quế20], 3., p. 89). Cho dãy $\{a_n\}_{n=0}^\infty$ được định nghĩa bởi công thức truy hồi:

$$a_0 = 1, a_n = f(n)a_0 + f(n-1)a_1 + f(n-2)a_2 + \cdots + f(1)a_{n-1}, \forall n \in \mathbb{N}^*,$$

với $f: \mathbb{N}^* \rightarrow \mathbb{R}$ là 1 hàm số được định nghĩa trước, e.g., lấy $f(n) = \frac{n^2 \sin n}{\sqrt{n(n+1)}|\cos n|}$, $\forall n \in \mathbb{N}^*$. Viết chương trình Python sử dụng đệ quy để tính a_n với $n \in \mathbb{N}$ được nhập vào từ bàn phím.

Bài toán 23 ([Quế20], 4., p. 89). Tính giai thừa kép $n!!$ rồi tính tổng $S = 1!! - 2!! + 3!! - 4!! + \cdots + (-1)^{n+1}n!!$ với $n \in \mathbb{N}^*$ được nhập vào từ bàn phím.

Bài toán 24 ([Quế20], 5., p. 90, Pascal triangle – tam giác Pascal). In lên màn hình tam giác Pascal chứa các hệ số của khai triển nhị thức $(x+1)^n$. Yêu cầu: Xuất lên màn hình tam giác Pascal “vuông cân” & tam giác Pascal “đều”.

Bài toán 25 ([Quế20], 6., p. 90). Viết chương trình Python sử dụng đệ quy để tính tổng: (a) $S = \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2n}$.

(b) $S = 1 + 2^2 + 3^3 + 4^4 + \cdots + n^n$. (c) $S = \sqrt{3n + \sqrt{3(n-1) + \cdots + \sqrt{9 + \sqrt{6 + \sqrt{3}}}}}$. (d) $S = \sqrt[n]{n} + \sqrt[n-1]{n-1} + \cdots + \sqrt{2}$.

(e) $S = \sqrt[n+1]{n + \sqrt[n]{n-1 + \cdots + \sqrt[3]{2 + \sqrt{1}}}}$. (f) $S = \sqrt{n + \sqrt{n-1 + \sqrt{n-2 + \cdots + \sqrt{2 + \sqrt{1}}}}}$.

(g) $S = \sqrt[n+1]{n! + \sqrt[n]{(n-1)! + \cdots + \sqrt[3]{2! + \sqrt{1!}}}}$. (h) $S(n) = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$ với $x \in \mathbb{R}$, rồi tính $e^x - S(n)$.

Bài toán 26 ([Quế20], 7., p. 91). Cho trước 1 tập gồm $n \in \mathbb{N}^*$ số nguyên. Liệt kê mọi cách phân hoạch tập này thành các tập con, e.g., tập $\{1, 2, 4\}$ có thể phân hoạch thành $\{1, 2, 4\} = \{1\} \cup \{2, 4\} = \{2\} \cup \{1, 4\} = \{4\} \cup \{1, 2\}$.

Bài toán 27 ([Quế20], 8., p. 91). Tính giá trị Ackerman $A(m, n)$ được định nghĩa bởi:

$$A(m, n) = \begin{cases} n+1 & \text{if } m = 0, \\ A(m-1, 1) & \text{if } n = 0, m > 0, \\ A(m-1, A(m, n-1)) & \text{if } m > 0, n > 0. \end{cases}$$

Với m, n nhập vào từ bàn phím, in ra ma trận $\{A(i, j)\}_{i,j=0}^{m,n}$.

Bài toán 28 ([Quế20], 9., p. 91). Viết chương trình Python sử dụng đệ quy để in ra tất cả dãy nhị phân dài n với $n \in \mathbb{N}^*$ được nhập vào từ bàn phím.

Bài toán 29 ([Quế20], 10., p. 91). Không được dùng list, tuple, string, chỉ dùng biến đơn, viết Python sử dụng đệ quy để đảo ngược 1 số, e.g., $123 \mapsto 321$.

3 Miscellaneous

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

Tài liệu

- [DT06] Lê Văn Doanh and Trần Khắc Tuấn. *101 Thuật Toán & Chương Trình Bài Toán Khoa Học Kỹ Thuật & Kinh Tế Bằng Ngôn Ngữ Turbo-Pascal*. In lần thứ 10. Nhà Xuất Bản Khoa Học & Kỹ Thuật, 2006, p. 268.
- [Đúc22] Nguyễn Tiến Đức. *Tuyển Tập 200 Bài Tập Lập Trình Bằng Ngôn Ngữ Python*. Nhà Xuất Bản Đại Học Thái Nguyên, 2022, p. 327.
- [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.
- [Quế20] Trần Thông Quế. *Bài Tập Lập Trình Cơ Bản Với Ngôn Ngữ Python*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2020, p. 194.
- [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.
- [Tru23] Vương Thành Trung. *Tuyển Tập Đề Thi Học Sinh Giỏi Cấp Tỉnh Trung Học Phổ Thông Tin Học*. Tài liệu lưu hành nội bộ, 2023, p. 235.
- [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.