

# Đề Thi Giữa Kỳ Nền Tảng Công Nghệ Thông Tin II 2025

## Information Technology Fundamentals II 2025

Nguyễn Quân Bá Hồng\*

Ngày 13 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/](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](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](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/lecture/NQBH_IT_fundamentals_lecture.tex).
- *Đề thi giữa kỳ môn Nền Tảng Công Nghệ Thông Tin II Hè 2025.*  
URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/resource/IFT2\\_midterm\\_exam\\_2025/NQBH\\_ITF2\\_midterm\\_exam\\_2025.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/resource/IFT2_midterm_exam_2025/NQBH_ITF2_midterm_exam_2025.pdf).  
TeX: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/resource/IFT2\\_midterm\\_exam\\_2025/NQBH\\_ITF2\\_midterm\\_exam\\_2025.tex](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/resource/IFT2_midterm_exam_2025/NQBH_ITF2_midterm_exam_2025.tex).
- (This file) *Lời giải đề thi giữa kỳ môn Nền Tảng Công Nghệ Thông Tin II Hè 2025.*  
URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/resource/IFT2\\_midterm\\_exam\\_2025/NQBH\\_ITF2\\_midterm\\_exam\\_2025\\_solution.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/resource/IFT2_midterm_exam_2025/NQBH_ITF2_midterm_exam_2025_solution.pdf).  
TeX: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/resource/IFT2\\_midterm\\_exam\\_2025/NQBH\\_ITF2\\_midterm\\_exam\\_2025\\_solution.tex](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/resource/IFT2_midterm_exam_2025/NQBH_ITF2_midterm_exam_2025_solution.tex).
- Codes:
  - Input: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/Python/ITF2\\_midterm\\_exam\\_2025\\_solution.inp](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/Python/ITF2_midterm_exam_2025_solution.inp).
  - Output: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/Python/ITF2\\_midterm\\_exam\\_2025\\_solution.out](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/Python/ITF2_midterm_exam_2025_solution.out).
  - Python: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/Python/ITF2\\_midterm\\_exam\\_2025\\_solution.py](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/Python/ITF2_midterm_exam_2025_solution.py).
- NGUYỄN LÊ ĐĂNG KHOA [NLDK]. *Nền Tảng Công Nghệ Thông Tin 2: Các Loại Dữ Liệu & Toán Tử Cơ Bản.*  
URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/IT\\_fundamentals/resource/SOT\\_Python\\_beginner.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/IT_fundamentals/resource/SOT_Python_beginner.pdf).

### Yêu cầu.

1. Viết đúng định dạng nhập xuất của file input và file output.
2. Được sử dụng tài liệu giấy không giới hạn số lượng.
3. Cấm sử dụng thiết bị điện tử, AIs. Nếu phát hiện 0 điểm ngay lần đầu tiên (không có cảnh cáo).
4. Thực hiện theo yêu cầu cụ thể của bài toán, e.g., nếu bài toán yêu cầu sử dụng vòng lặp `for` thì phải sử dụng vòng lặp `for`, không sử dụng vòng lặp `while`.
5. Các chú thích code (comments code) phải được đặt sau dấu `#` hoặc giữa `''' <comment block> '''` nếu dùng khối chú thích code.

---

\*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](mailto:nguyenquanbahong@gmail.com) & [hong.nguyenquanba@umt.edu.vn](mailto:hong.nguyenquanba@umt.edu.vn). Website: <https://nqbh.github.io/>. GitHub: <https://github.com/NQBH>.

6. Nếu sử dụng hàm có sẵn (built-in function) nào của Python, thì phải import thư viện tương ứng, e.g., `import math`, nếu sử dụng hàm `sqrt` của thư viện `math` thì khai báo `from math import sqrt`.
7. Nếu không làm được ý trước, vẫn có thể sử dụng kết quả các ý trước để làm ý sau của bài toán.
8. Đề có tổng điểm là 13, nếu kết quả hơn 10, thì cộng phần điểm dư vào các cột khác.

**Bài 1** (Tính chu vi & diện tích hình chữ nhật). (1.5 điểm) *Viết chương trình Python để tính chu vi  $P$ , diện tích  $S$ , & đường chéo  $d$  của hình chữ nhật với 2 kích thước  $a, b \in (0, \infty)$  được nhập vào.*

**Input.** File `rectangle.inp` chỉ gồm 1 dòng chứa 2 số thực  $a, b$  (phải kiểm tra điều kiện dương).

**Output.** In ra chu vi của hình chữ nhật theo công thức  $P = 2(a + b)$ , diện tích của hình chữ nhật theo công thức  $S = ab$ , đường chéo của hình chữ nhật theo công thức  $d = \sqrt{a^2 + b^2}$ .

**Sample.**

rectangle.inp	rectangle.out
4.5 6	$P = 21$ $S = 27$ $d = 7.5$

*Solution.* C++:

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      double a, b;
7      cin >> a >> b;
8      if (a <= 0 or b <= 0)
9          cout << "Error: a & b must be positive.";
10     else
11         cout << "P = " << 2 * (a + b) << '\n' << "S = " << a * b << '\n' << "d = " << sqrt(a * a + b * b);
12 }
```

Python:

```

1  a, b = map(float, input().split())
2  if a <= 0 or b <= 0:
3      print('Error: a & b must be positive.')
4  else:
5      print('P =', 2 * (a + b))
6      print('S =', a * b)
7      print('d =', sqrt(a * a + b * b) ** 1 / 2)
```

□

**Bài 2** (Tính giá trị biểu thức/hàm số). (1.5 điểm) *Viết hàm Python với tên `def evaluate_function(a, b, c):` để tính trực tiếp, không dùng hàm, giá trị của hàm số*

$$A(a, b, c) = \frac{a^4 b^3 \sqrt[3]{c}}{\sqrt{a - 2\sqrt[3]{b}(c^2 - 1)}}$$

với 3 số thực  $a, b, c \in \mathbb{R}$  được nhập từ bàn phím hoặc file input.

**Input.** File `evaluate_function.inp` chứa 3 số thực  $a, b, c$  (phải kiểm tra điều kiện để biểu thức  $A(a, b, c)$  xác định).

**Output.** Nếu có lỗi, in ra lỗi chia cho 0, lỗi lấy căn bậc chẵn của số thực âm tương ứng. Nếu không có lỗi, in ra giá trị của biểu thức  $A(a, b, c)$ .

**Sample.**

evaluate_function.inp	evaluate_function.out
2 1 3	Error: division by 0.
1 1 2	Error: square root of negative real number.
3 1 2	34.017868347161574

*Solution.* Phân tích: Biểu thức  $A(a, b, c)$  chỉ xác định iff  $a > 2, b \neq 0, c \neq \pm 1$ , khi đó hàm số  $A : (2, \infty) \times \mathbb{R}^* \times \mathbb{R} \setminus \{\pm 1\}$  xác định & có thể được tính như sau:

C++:

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      double a, b, c;
7      cin >> a >> b >> c;
8      if (a < 2)
9          cout << "Error: square root of negative real number.";
10     else if (a == 2 or b == 0 or c == -1 or c == 1)
11         cout << "Error: division by 0.";
12     else
13         cout << pow(a, 4) * pow(b, 3 - 1.0 / 3) * pow(c, 1.0 / 3) * 1.0 / (sqrt(a - 2) * (c * c - 1));
14 }
```

Python:

```

1  a, b, c = map(float, input().split())
2  if a < 2:
3      print('Error: square root of negative real number.')
4  elif a == 2 or b == 0 or c == -1 or c == 1:
5      print('Error: division by 0.')
6  else:
7      print(f'A({a}, {b}, {c}) =', (a ** 4 * b ** (3 - 1/3) * c ** (1/3)) / (sqrt(a - 2) * (c ** 2 - 1)))
```

□

**Bài 3** (Tổng chứa giai thừa). (2 điểm) Dùng vòng lặp for, viết chương trình Python để tính tổng

$$S!(n) = \sum_{i=1}^n i \cdot i! = 1 \cdot 1! + 2 \cdot 2! + \cdots + (n-1)(n-1)! + n \cdot n!,$$

với  $n \in \mathbb{N}^*$  được nhập vào rồi so sánh kết quả với  $f(n) = (n+1)! - 1$ .

Input. File sum.inp chỉ gồm 1 số nguyên dương  $n \in \mathbb{N}^*$ .

Output. Xuất ra giá trị  $S!(n)$  rồi kết quả so sánh  $S!(n)$  với  $f(n)$  (<, > hay =).

Sample.

sum_fact.inp	sum_fact.out
2	5 $S!(2) = f(2) = 5$
3	23 $S!(3) = f(3) = 23$

*Solution.* C++:

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8      if (n <= 0)
9          cout << "Error: n must be a positive integer.";
10     else {
11         long long S = 1, factorial = 1;
```

```

12         for (int i = 2; i <= n; ++i) {
13             factorial *= i;
14             S += i * factorial;
15         }
16         cout << "S! = " << S << '\n';
17         if (S == factorial * (n + 1) - 1)
18             cout << "S!(" << n << ") = f(" << n << ") = " << S;
19         else if (S < factorial * (n + 1) - 1)
20             cout << "S!(" << n << ") < f(" << n << ") = " << S;
21         else if (S > factorial * (n + 1) - 1)
22             cout << "S!(" << n << ") > f(" << n << ") = " << S;
23     }
24 }

```

Python:

```

1  n = int(input())
2  if n <= 0:
3      print("Error: n must be a positive integer.")
4  else:
5      S = 1
6      factorial = 1
7      for i in range(2, n + 1):
8          factorial *= i
9          S += i * factorial
10     print("S! =", S)
11     if S == factorial * (n + 1) - 1:
12         print(f'S!{n} = f({n}) =', S)

```

hoặc sử dụng hàm `math.factorial`:

```

1  from math import factorial
2  n = int(input())
3  if n <= 0:
4      print("Error: n must be a positive integer.")
5  else:
6      S = 1
7      factorial = 1
8      for i in range(2, n + 1):
9          S += i * factorial(i)
10     print("S! =", S)
11     if S == factorial * (n + 1) - 1:
12         print(f'S!{n} = f({n}) =', S)

```

□

**Bài 4** (Chỉ số nhỏ nhất). (2 điểm) Dùng vòng lặp `while`, viết chương trình Python để số  $n \in \mathbb{N}^*$  nhỏ nhất thỏa mãn

$$S(n) = \sum_{i=1}^n i \cdot i! = 1 \cdot 1! + 2 \cdot 2! + \cdots + (n-1)(n-1)! + n \cdot n! > a,$$

với số thực  $a \in \mathbb{R}$  được nhập vào.

Input. File input gồm 1 số thực  $a \in \mathbb{R}$ .

Output. Xuất ra chỉ số nhỏ nhất thỏa mãn  $S(n) > a$ .

Sample.

min_index.inp	min_index.out
4.97	2
22.99	3

*Solution.* Python:

```
1 a = float(input())
2 i = S = factorial = 1
3 while S < a:
4     i += 1
5     factorial *= i
6     S += i * factorial
7 print(i)
```

□

**Bài 5** (Bao phủ hình chữ nhật nguyên 2D nhỏ nhất). (2.5 điểm) *Viết chương trình Python để tính chu vi, diện tích, & độ dài đường chéo hình chữ nhật nhỏ nhất “chứa trọn”  $n \in \mathbb{N}^*$  điểm  $A_1(x_1, y_1), A_2(x_2, y_2), \dots, A_n(x_n, y_n) \in \mathbb{R}^2$  với tọa độ nguyên  $x_i, y_i \in \mathbb{Z}, \forall i = 1, \dots, n$ , cho trước trong mặt phẳng 2 chiều, “chứa trọn” ở đây nghĩa là các điểm chỉ được nằm bên trong hình chữ nhật, không được nằm trên cạnh hình chữ nhật.*

**Input.** Dòng 1 chứa số nguyên  $n \in \mathbb{N}^*$ : \* số điểm trong mặt phẳng 2 chiều.  $n$  dòng tiếp theo, mỗi dòng chứa 2 số nguyên  $x_i, y_i \in \mathbb{Z}$ : hoành độ & tung độ của điểm thứ  $i$   $A_i(x_i, y_i)$ .

**Output.** In ra chu vi, diện tích, & đường chéo của hình chữ nhật thỏa mãn nhờ gọi lại hàm của Bài 1 (hoặc tự viết lại nếu muốn).

Sample.

min_rectangle.inp	min_rectangle.out
4	$P = 26$
1 0	$S = 40$
-2 2	$d = 9.43398113206$
-1 3	
4 2	

*Solution.* Python:

```
1 n = int(input()) # number of 2D points -- số điểm trên mặt phẳng
2 x_min = y_min = 1e9
3 x_max = y_max = -1e9
4 for i in range(n):
5     x, y = map(int, input().split())
6     if x < x_min:
7         x_min = x
8     if x > x_max:
9         x_max = x
10    if y < y_min:
11        y_min = y
12    if y > y_max:
13        y_max = y
14 a, b = x_max - x_min + 2, y_max - y_min + 2
15 print('P =', 2 * (a + b))
16 print('S =', a * b)
17 print('d =', sqrt(a * a + b * b))
```

□

**Bài 6** (Bao phủ hình hộp chữ nhật nguyên 3D nhỏ nhất). (3.5 điểm) *Viết chương trình Python để tính diện tích toàn phần, thể tích, & độ dài đường chéo hình hộp chữ nhật nhỏ nhất chứa  $n \in \mathbb{N}^*$  điểm  $A_1(x_1, y_1, z_1), A_2(x_2, y_2, z_2), \dots, A_n(x_n, y_n, z_n) \in \mathbb{R}^3$  với tọa độ nguyên  $x_i, y_i, z_i \in \mathbb{Z}, \forall i = 1, \dots, n$ , cho trước trong không gian 3 chiều, ở đây các điểm có thể nằm bên trong hoặc nằm trên cạnh hình hộp chữ nhật.*

**Input.** Dòng 1 chứa số nguyên  $n \in \mathbb{N}^*$ : số điểm trong không gian 3 chiều.  $n$  dòng tiếp theo, mỗi dòng chứa 3 số nguyên  $x_i, y_i, z_i \in \mathbb{Z}$ : hoành độ, tung độ, & cao độ của điểm thứ  $i$   $A_i(x_i, y_i, z_i)$ .

**Output.** In ra diện tích toàn phần, thể tích, & độ dài đường chéo hình hộp chữ nhật thỏa mãn, biết với hình hộp chữ nhật có kích thước  $a \times b \times c$  thì  $S_{tp} = 2(ab + bc + ca), V = abc, d = \sqrt{a^2 + b^2 + c^2}$ .

Sample.

min_rectangular_cuboid.inp	min_rectangular_cuboid.out
4 1 0 0 0 1 0 -1 0 2 2 0 1	$S_{tp} = 22$ $V = 6$ $d = 3.74165738677$

*Solution.* Python:

```
1  n = int(input()) # number of 3D points
2  x_min = y_min = z_min = 1e9
3  x_max = y_max = z_max = -1e9
4  for i in range(n):
5      x, y, z = map(int, input().split())
6      if x < x_min:
7          x_min = x
8      if x > x_max:
9          x_max = x
10     if y < y_min:
11         y_min = y
12     if y > y_max:
13         y_max = y
14     if z < z_min:
15         z_min = z
16     if z > z_max:
17         z_max = z
18 a, b, c = x_max - x_min, y_max - y_min, z_max - z_min
19 print('S_tp =', 2 * (a * b + b * c + c * a))
20 print('V =', a * b * c)
21 print('d =', sqrt(a * a + b * b + c * c))
```

□