



ĐẶNG TUẤN THÀNH



NHẬP MÔN LẬP TRÌNH CƠ BẢN



Yên Bái, Tháng 9 năm 2022



LỜI NÓI ĐẦU

Nhằm đáp ứng yêu cầu cần đạt của Chủ đề F: Giải quyết vấn đề với sự trợ giúp của máy tính trong môn Tin học 10 theo Chương trình phổ thông 2018 và chương trình môn Tin học lớp 11. Nhóm tác giả đã nghiên cứu và biên soạn quyển sách “**Nhập môn Lập trình cơ bản**”.

Nội dung sách gồm **110** bài tập được chia thành năm phần:

- 🚩 **Phần 1. Cấu trúc tuần tự**
- 🚩 **Phần 2. Cấu trúc rẽ nhánh**
- 🚩 **Phần 3. Cấu trúc lặp**
- 🚩 **Phần 4. Kiểu mảng**
- 🚩 **Phần 5. Kiểu xâu**

Tác giả đã biên tập tỉ mỉ từng bài tập nhằm giúp người đọc khai thác cuốn sách một cách hiệu quả. Mỗi bài tập được cấu trúc gồm:

- ✓ Đề bài, test mẫu;
- ✓ Hướng dẫn thuật toán;
- ✓ Mã chương trình viết bởi ba ngôn ngữ lập trình là ***Pascal, C++ và Python***.
- ✓ Test để chấm tự động.

Từ đó người đọc từng bước thành kiến thức, kỹ năng và làm chủ được ngôn ngữ lập trình Python. Đi kèm quyển sách là diễn đàn học lập trình do tác giả xây dựng <https://antit.forumvi.com/>.

Trong thời gian tới, tác giả viết tiếp quyển sách “***Phương pháp sinh (generation)***” với các bài tập chuyên đề mới và có mã chương trình được viết bởi ba ngôn ngữ lập trình (Pascal, C++ và Python). Rất mong các bạn đón đọc.

Tác giả xin trân trọng, biết ơn em Lê Quang Lâm, em Nguyễn Trần Thế Anh, em Nguyễn Nhật Quân, em Nguyễn Phúc Tâm và em Đỗ Hoàng Phúc đã giúp tác giả hoàn thành quyển sách này.

Quyển sách này được viết trong thời gian gấp rút, không tránh được những thiếu sót, tác giả rất mong được sự đóng góp nội dung, phản hồi của bạn đọc vào email: dtthanh.c3ntt@yenbai.edu.vn.

Tác giả

Mục lục

LỜI NÓI ĐẦU.....	3
PHẦN 1. CẤU TRÚC TUẦN TỰ.....	8
Bài 1. Độ dài đoạn thẳng 1	8
Bài 2. Chu vi và diện tích tam giác	9
Bài 3. Độ dài đoạn thẳng 2	10
Bài 4. Lát gạch 1	12
Bài 5. Tổng các số tự nhiên liên tiếp.....	13
Bài 6. Tổng bình phương các số tự nhiên liên tiếp	14
Bài 7. Tổng các số lẻ liên tiếp.....	15
Bài 8. Tổng lập phương các số tự nhiên liên tiếp	15
Bài 9. Tổng lũy thừa 5 các số tự nhiên liên tiếp	16
Bài 10. Tổng cặp số tự nhiên nhân dồn.....	17
Bài 11. Tổng các nghịch đảo cặp số tự nhiên nhân dồn.....	17
Bài 12. Tổng các nghịch đảo cặp ba số tự nhiên nhân dồn	18
Bài 13. Tổng dãy đặc biệt 1	19
Bài 14. Tổng dãy đặc biệt 2	20
Bài 15. Tổng dãy đặc biệt 3	21
PHẦN 2. CẤU TRÚC RỄ NHÁNH.....	23
Bài 16. Chẵn lẻ.....	23
Bài 17. Chia hết cho 3	23
Bài 18. Giá trị lớn nhất.....	24
Bài 19. Giá trị nhỏ nhất	25
Bài 20. Cặp số có tổng lớn nhất.....	26
Bài 21. Ghép số.....	27
Bài 22. Giải phương trình bậc nhất.....	28
Bài 23. Giải phương trình bậc 2.....	29
Bài 24. Tam giác 1	32
Bài 25. Điểm và đường tròn	34
Bài 26. Tam giác 2	35
Bài 27. Số chẵn lớn nhất.....	37

Bài 28. Chữ số lớn nhất.....	38
Bài 29. Hoán vị chữ số lớn nhất.....	40
Bài 30. Xóa số.....	42
Bài 31. Giải phương trình 3.....	44
Bài 32. Điểm trong hình chữ nhật.....	46
Bài 33. Điểm trong tam giác.....	47
PHẦN 3. CẤU TRÚC LẶP.....	51
I. Câu lệnh lặp For.....	51
Bài 34. Tổng các phân số liên tiếp.....	51
Bài 35. Tổng các phân số lẻ liên tiếp.....	52
Bài 36. Giai thừa của N.....	53
Bài 37. Chia hết cho 2 và không chia hết cho 3.....	55
Bài 38. Đếm số chính phương.....	55
Bài 39. Liệt kê ước nguyên dương.....	57
Bài 40. Mua hàng.....	57
Bài 41. Đổi tiền 1.....	58
Bài 42. Đổi tiền 2.....	60
Bài 43. Đếm số lượng ước.....	62
Bài 44. Ước chẵn.....	62
Bài 45. Tổng ước.....	63
Bài 46. Số nguyên tố.....	63
Bài 47. Liệt kê số nguyên tố.....	64
II. Câu lệnh lặp while.....	65
Bài 48. Tính S.....	65
Bài 49. Ước chung lớn nhất.....	66
Bài 50. Thừa số nguyên tố.....	66
Bài 51. Số nguyên tố 2.....	67
Bài 52. Tổng chữ số của N.....	67
Bài 53. Cắt số.....	68
Bài 54. Ước nguyên tố lớn nhất.....	69
PHẦN 4. KIỂU MẢNG.....	69

Bài 55. Mảng một chiều	69
Bài 56. Liệt kê phần tử là số chẵn.....	70
Bài 57. Vị trí phần tử là số chính phương	71
Bài 58. Đếm phần tử.....	72
Bài 59. Tìm phần tử nhỏ nhất	74
Bài 60. Đếm số lượng phần tử lớn nhất	75
Bài 61. Tính tổng và trung bình cộng	77
Bài 62. Trung bình cộng 2	78
Bài 63. Vị trí K.....	80
Bài 64. Chênh lệch lớn nhất.....	81
Bài 65. Tổng lớn nhất.....	83
Bài 66. Dãy không giảm.....	84
Bài 67. Dãy Fibonacci	86
Bài 68. Dãy Fibonacci nguyên tố.....	88
Bài 69. Dãy đảo.....	88
Bài 70. Nhân đôi dãy.....	89
Bài 71. Mảng hai chiều.....	90
Bài 72. Tổng số chẵn.....	91
Bài 73. Đếm số lượng.....	92
Bài 74. Vị trí phần tử chia hết cho K	93
Bài 75. Tìm giá trị lớn nhất	95
Bài 76. Đếm số lượng giá trị nhỏ nhất	96
Bài 77. Tìm giá trị lớn nhất của mỗi hàng.....	97
Bài 78. Tìm giá trị nhỏ nhất của mỗi cột.....	99
Bài 79. Tìm tổng từng hàng, từng cột	100
Bài 80. Vị trí các số chính phương	102
Bài 81. Vị trí các số nguyên tố.....	103
PHẦN 5. KIỂU XÂU.....	104
Bài 82. Xâu 1	104
Bài 83. Vị trí S1 trong S	104
Bài 84. Đếm xâu S1 trong xâu S.....	105

Bài 85. Nối xâu	107
Bài 86. Đếm kí tự số.....	108
Bài 87. Xâu đảo.....	109
Bài 88. Xâu đối xứng	111
Bài 89. Xâu chữ hoa.....	112
Bài 90. Xâu chữ thường.....	113
Bài 91. Chuẩn hóa xâu	114
Bài 92. Đếm số lượng từ trong xâu	115
Bài 93. Kí tự đầu tiên các từ in hoa.....	117
Bài 94. Đếm phân phối.....	118
Bài 95. Tách số trong xâu	121
Bài 96. Số lớn nhất trong xâu	122
Bài 97. Sắp xếp kí tự	124
Bài 98. Đếm kí tự 1	125
Bài 99. Nén xâu.....	126
Bài 100. Giải nén xâu.....	128
Bài 101. Xóa S1 trong S	130
Bài 102. Xâu Fibonacci 1	131
Bài 103. Xâu Fibonacci 2	132
Bài 104. Xâu con palidrome	133
Bài 105. Sắp xếp số trong xâu	134
Bài 106. Mã hóa xâu	137
Bài 107. Giải mã xâu.....	138
Bài 108. Ghép xâu số	140
Bài 109. Xâu con chung dài nhất.....	141
Bài 110. Nhầm lẫn.....	142

PHẦN 1. CẤU TRÚC TUẦN TỰ

Bài 1. Độ dài đoạn thẳng 1

Cho $A(x_1, y_1)$, $B(x_2, y_2)$. Tính độ dài đoạn thẳng AB. Kết quả lấy chính xác đến 2 chữ số thập phân.

Ví dụ: $A(0, 0)$, $B(2, 0)$. Kết quả: $d_{AB} = 2.00$

BAI1.INP	BAI1.INP
0	2.00
0	
2	
0	

❖ Thuật toán:

Áp dụng công thức tính độ dài đoạn thẳng khi biết tọa độ các điểm:

$$d_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Để tính bình phương ta dùng hàm **pow**(a, b) = ab

Để tính căn bậc hai ta dùng hàm **sqrt**(x) = \sqrt{x}

Gọi hai hàm này trong thư viện **math** ra:

```
from math import sqrt
from math import pow
```

❖ Chương trình tham khảo:

Code Python	Code Pascal
<pre>from math import sqrt from math import pow x1 = float(input("Enter x1: ")) y1 = float(input("Enter y1: ")) x2 = float(input("Enter x2: ")) y2 = float(input("Enter y2: ")) res = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2)) print(res)</pre>	<pre>var x1,x2,y1,y2,z1,z2:longint; d:real; begin readln(x1,x2,y1,y2,z1,z2); d:=sqrt(sqr((x2-x1))+sqr((y2- y1))+sqr((z2-z1))); writeln(d:0:3); end.</pre>

Code C++	

Bài 2. Chu vi và diện tích tam giác

Cho a, b, c là ba cạnh của tam giác ABC. Hãy tính chu vi và diện tích tam giác ABC. Kết quả lấy chính xác đến 3 chữ số thập phân.

Ví dụ: a = 3, b = 4, c = 5. Kết quả: cv = 12.000; dt = 6.00.

BAI2.INP	BAI2.INP
3	6.00
4	
5	

❖ Thuật toán:

Áp dụng công thức Hê-rông để tính diện tích tam giác khi biết độ dài các cạnh.

Gọi P là nửa chu vi thì $p = \frac{a+b+c}{2}$; diện tích $s = \sqrt{p(p-a)(p-b)(p-c)}$

Ta gọi hàm **sqrt(x)** từ thư viện **math**: **from math import sqrt**

❖ Video bài giảng: https://youtu.be/_9T8qAmBUro

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <math.h> #include <stdio> using namespace std; float a, b, c, p, s; int main() { cin >> a >> b >> c; p = a + b + c;</pre>	<pre>from math import sqrt a = int(input("Enter a: ")) b = int(input("Enter b: ")) c = int(input("Enter c: ")) d = a + b + c p = d / 2.0 s = sqrt(p * (p - a) * (p - b) * (p - c)) print(d) print(s)</pre>
--	--

<pre> s = sqrt((p / 2.0) * (p / 2.0 - a) * (p / 2.0 - b) * (p / 2.0 - c)); printf("%0.2f %0.2f", p, s); return 0; } </pre>	
<pre> var a,b,c:longint; s,d:real; begin s:=0; d:=0; readln(a,b,c); d:=(a+b+c) div 2; writeln(d*2:0:3); s:=sqrt(d*(d-a)*(d- b)*(d-c)); writeln(s:0:3); end. </pre>	

Bài 3. Độ dài đoạn thẳng 2

Cho A(x1, y1), B(x2, y2), C(x3, y3) là ba đỉnh của tam giác ABC. Hãy tính độ dài các đoạn thẳng dAB, dBC, dCA. Kết quả lấy chính xác đến 3 chữ số thập phân.

❖ Thuật toán:

Áp dụng công thức tính độ dài đoạn thẳng khi biết tọa độ các điểm:

$$d_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Vì cần áp dụng công thức nhiều lần với các bộ số (x, y) khác nhau để tính độ dài các đoạn thẳng nên ta tổ chức chương trình con dạng hàm: **float dodai(int xM, int yM, int xN, int yN).**

Khi đó, ta có:

```
dab = dodai(xa, ya, xb, yb); dbc = dodai(xb, yb, xc, yc);
```

```
dca = dodai(xc, yc, xa, ya);
```

Để viết ra có quy cách trong Python ta dùng lệnh: **print("c={:0.2f}".format(c))**, với c là giá trị cần viết ra lấy chính xác 2 chữ số thập phân.

Ta định nghĩa hàm bằng từ khoá **def**

```
def dodai(a1,b1,a2,b2):
    a = a2 - a1
    b = b2 - b1
    c = sqrt(pow(a,2) + pow(b,2))
    return c
```

❖ Video bài giảng: <https://youtu.be/YYOESmM4mvg>

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <math.h> #include <stdio> using namespace std; int xa, ya, xb, yb, xc, yc; float dab, dbc, dca; float dodai(int xm, int ym, int xn, int yn) { float d = sqrt((xm - xn) * (xm - xn) + (ym - yn) * (ym - yn)); return d; } int main() { cin >> xa >> ya >> xb >> yb >> xc >> yc; dab = dodai(xa, ya, xb, yb); dbc = dodai(xb, yb, xc, yc); dca = dodai(xc, yc, xa, ya);</pre>	<pre>from math import sqrt from math import pow def distance_points(ax, ay, bx, by): return sqrt(pow(bx - ax, 2) + pow(by - ay, 2)) x1 = float(input("Enter x1: ")) y1 = float(input("Enter y1: ")) x2 = float(input("Enter x2: ")) y2 = float(input("Enter y2: ")) x3 = float(input("Enter x3: ")) y3 = float(input("Enter y3: ")) dab = distance_points(x1, y1, x2, y2) dbc = distance_points(x2, y2, x3, y3) dac = distance_points(x1, y1, x3, y3) print("d_AB = {:.3f}".format(dab)) print("d_BC = {:.3f}".format(dbc)) print("d_AC = {:.3f}".format(dac))</pre>
--	--

<pre> printf("%0.3f %0.3f %0.3f", dab, dbc, dca); return 0; } </pre>	
<pre> var x1,x2,y1,y2,z1,z2:longint; d1,d2,d3:real; begin readln(x1,x2,y1,y2,z1,z2); d1:=sqrt(sqr((x2-x1))+sqr((y2-y1))); writeln(d1:0:3); d2:=sqrt(sqr((y2-y1))+sqr((z2-z1))); writeln(d2:0:3); d3:=sqrt(sqr((x2-x1))+sqr((z2-z1))); writeln(d3:0:3); end. </pre>	

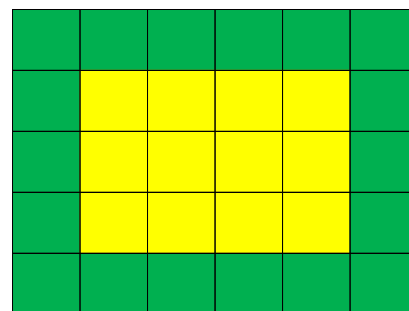
Bài 4. Lát gạch 1

Cho hình chữ nhật kích thước $a \times b$ được chia thành lưới ô vuông đơn vị 1×1 . Người ta lát hình chữ nhật bằng những viên gạch kích thước 1×1 màu xanh và màu vàng. Màu xanh được lát ở viền, màu vàng được lát ở phần lõi còn lại.

Hãy tính số viên gạch màu xanh và viên gạch màu vàng.

Ví dụ: $a = 5, b = 6$

Kết quả: 18 12



❖ Thuật toán:

Gọi y là số viên gạch màu vàng, $y = (a - 2)(b - 2)$

Gọi x là số viên gạch màu xanh, $x = a * b - y$.

Ta đưa x, y ra.

❖ Video bài giảng: <https://youtu.be/Mgk8drnaArw>

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int a, b, x, y; int main() { cin >> a >> b; y = (a - 2) * (b - 2); x = a * b - y; cout << x << " " << y; return 0; }</pre>	<pre>a = int(input("Nhap a: ")) b = int(input("Nhap b: ")) x = (a - 2) * (b - 2) y = a * b - x print(x) print(y)</pre>

Bài 5. Tổng các số tự nhiên liên tiếp

Cho N nguyên dương. Hãy tính $S = 1 + 2 + 3 + \dots + N$.

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1 + 2 + 3 + \dots + N = \frac{N(N + 1)}{2}$$

Lưu ý, tổng S có thể rất lớn nên ta dùng kiểu nguyên **long long**, đồng thời ép kiểu dữ liệu của vế phải thành kiểu long long bằng cách nhân biểu thức với hằng số *1LL*.

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int n; long long s;</pre>	<pre>var i,n,s:longint; begin readln(n); s:=0; for i:=1 to n do s:=s+i;</pre>
---	---

<pre>int main() { cin >> n; s = n*1LL * (n + 1) / 2; cout << s << endl; return 0; }</pre>	<pre>write(s); end.</pre>
<pre># Sum of consecutive integers calculation program n = int(input("Nhap n: ")) s = n * (n + 1) // 2 print("Result: ", s)</pre>	

Bài 6. Tổng bình phương các số tự nhiên liên tiếp

Cho N nguyên dương. Hãy tính $S = 1^2 + 2^2 + 3^2 + \dots + N^2$.

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1^2 + 2^2 + 3^2 + \dots + N^2 = \frac{N.(N+1)(2N+1)}{6}.$$

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int n; long long s; int main() { cin >> n; s = n*1LL * (n + 1) * (2 * n + 1) / 6; cout << s << endl; return 0; }</pre>	<pre>var s,i,n:longint; begin readln(n); s:=0; for i:=1 to n do s:=s+sqr(i); write(s); end.</pre>
---	---

<pre> n = int(input("Nhap n: ")) s = n * (n + 1) * (2 * n + 1) // 6 print("Result: ", s) </pre>	

Bài 7. Tổng các số lẻ liên tiếp

Cho N nguyên dương. Hãy tính $S = 1 + 3 + 5 + \dots + (2 * N - 1)$.

❖ **Thuật toán:**

Áp dụng công thức tính tổng các số lẻ liên tiếp:

$$S = 1 + 3 + 5 + \dots + (2 * N - 1) = N^2.$$

❖ **Chương trình tham khảo:**

<pre> #include <iostream> using namespace std; int n; long long s; int main() { cin >> n; s = 1LL * n * n; cout << s << endl; return 0; } </pre>	<pre> n = int(input("Nhap n: ")) print(n * n) </pre>

Bài 8. Tổng lập phương các số tự nhiên liên tiếp

Cho N nguyên dương. Hãy tính $S = 1^3 + 2^3 + 3^3 + \dots + N^3$.

❖ **Thuật toán:**

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1^3 + 2^3 + 3^3 + \dots + N^3 = \left(\frac{N(N+1)}{2} \right)^2.$$

❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int n; long long s; int main() { cin >> n; s = 1LL * n * (n + 1) / 2; cout << s * s; return 0; }</pre>	<pre>n = int(input("Nhap n: ")) s = n * (n + 1) // 2 print(s * s)</pre>

Bài 9. Tổng lũy thừa 5 các số tự nhiên liên tiếp

Cho N nguyên dương. Hãy tính $S = 1^5 + 2^5 + 3^5 + \dots + N^5$.

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1^5 + 2^5 + 3^5 + \dots + N^5 = \frac{1}{12} N^2 (N + 1)^2 (2N^2 + 2N - 1).$$

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int n; long long s; int main() { cin >> n; s = 1LL * n * n * (n + 1) * (n + 1) * (2 * n * n + 2 * n - 1) / 12;</pre>	<pre>n = int(input("Nhap n: ")) s = n * n * (n + 1) * (n + 1) * (2 * n * n + 2 * n - 1) // 12 print(s)</pre>
---	--

<pre>cout << s; return 0; }</pre>	

Bài 10. Tổng cặp số tự nhiên nhân dồn

Cho N nguyên dương. Hãy tính $S = 1.2 + 2.3 + 3.4 + \dots + N(N + 1)$.

❖ Thuật toán:

Áp dụng công thức tính tổng:

$$S = 1.2 + 2.3 + 3.4 + \dots + N(N + 1) = \frac{N.(N+1).(N+2)}{3}.$$

Lưu ý, tổng S có thể rất lớn nên ta dùng kiểu nguyên **long long**, đồng thời ép kiểu dữ liệu của v phải thành kiểu long long bằng cách nhân biểu thức với hằng số **1LL**.

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int n; long long s; int main() { cin >> n; s = 1LL * n * (n + 1) * (n + 2) / 3; cout << s; return 0; }</pre>	<pre>n = int(input("Nhap n: ")) s = n * (n + 1) * (n + 2) // 3 print(s)</pre>

Bài 11. Tổng các nghịch đảo cặp số tự nhiên nhân dồn

Cho N nguyên dương.

Hãy tính $S = 1/(1.2) + 1/(2.3) + 1/(3.4) + \dots + 1/(N(N + 1))$.

❖ **Thuật toán:**

Áp dụng công thức tính tổng:

$$S = 1/(1.2) + 1/(2.3) + 1/(3.4) + \dots + 1/(N(N+1)) = \frac{N}{N+1}.$$

Lưu ý, phép chia / trong C++ của các số kiểu **int** kết quả trả về kiểu **int**. Vì vậy, cần ép tử số hoặc mẫu số sang kiểu số thực bằng cách nhân với hằng số **1.0**

❖ **Video bài giảng:** <https://youtu.be/TL2NU9nUwI8>❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int n; float s; int main() { cin >> n; s = n*1.0 / (n+1); cout << s; return 0; }</pre>	<pre>n = int(input("Nhập n: ")) s = n / (n + 1.0) print(s)</pre>

Bài 12. Tổng các nghịch đảo cặp ba số tự nhiên nhân dồn

Cho N nguyên dương.

Hãy tính $S = 1/(1.2.3) + 1/(2.3.4) + 1/(3.4.5) + \dots + 1/(N(N+1)(N+2))$

❖ **Thuật toán:**

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1/(1.2.3) + 1/(2.3.4) + 1/(3.4.5) + \dots + 1/(N(N+1)(N+2)) = \frac{N(N+3)}{4(N+1)(N+2)}.$$

Lưu ý, phép chia / trong C++ của các số kiểu **int** kết quả trả về kiểu **int**. Vì vậy, cần ép tử số hoặc mẫu số sang kiểu số thực bằng cách nhân với hằng số **1.0**

❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int n; float s; int main() { cin >> n; s = 1.0 * n * (n + 3) / (4.0 * (n + 1) * (n + 2)); cout << s; return 0; }</pre>	<pre>n = int(input("Nhap n: ")) s = n * (n + 3) / (4.0 * (n + 1) * (n + 2)) print(s)</pre>

Bài 13. Tổng dãy đặc biệt 1

Cho N nguyên dương.

Hãy tính $S = \frac{3}{(1.2)^2} + \frac{5}{(2.3)^2} + \dots + \frac{2N+1}{(N.(N+1))^2}$

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = \frac{3}{(1.2)^2} + \frac{5}{(2.3)^2} + \dots + \frac{2N+1}{(N.(N+1))^2} = \frac{N(N+2)}{(N+1)^2}$$

Lưu ý, phép chia / trong C++ của các số kiểu int kết quả trả về kiểu int. Vì vậy, cần ép tử số hoặc mẫu số sang kiểu số thực bằng cách nhân với hằng số **1.0**

❖ Chương trình tham khảo:

<pre>#include <iostream> using namespace std; int n;</pre>	<pre># Tong day so dac biet 1 n = int(input("Nhap n: ")) s = n * (n + 2) / ((n + 1) ** 2) print(s)</pre>
--	--

<pre>float s; int main() { cin >> n; s = 1.0f * n * (n + 2.0f) / ((n + 1.0f) * (n + 1.0f)); cout << s; return 0; }</pre>	

Bài 14. Tổng dãy đặc biệt 2

Cho N và P nguyên dương.

Hãy tính $S = 1 + p^1 + p^2 + \dots + p^N$

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1 + p^1 + p^2 + \dots + p^N = \frac{p^{N+1} - 1}{p - 1}$$

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <math.h> using namespace std; int n; int p; long long s; int main() { cin >> n >> p;</pre>	<pre># Tong day dac biet 2 p = int(input("Nhap p: ")) n = int(input("Nhap n: ")) s = (p ** (n + 1) - 1) // (p - 1) print(s)</pre>
---	---

<pre> s = ((long long) pow(p, n + 1) - 1) / (p - 1); cout << s; return 0; } </pre>	

Bài 15. Tổng dãy đặc biệt 3

Cho N và P nguyên dương.

Hãy tính $S = 1 + 2p^1 + 3p^2 + \dots + (N + 1)p^N$

❖ Thuật toán:

Áp dụng công thức tính tổng các số tự nhiên liên tiếp:

$$S = 1 + 2p^1 + 3p^2 + \dots + (N + 1)p^N = \frac{(n + 1)p^{N+1}}{p - 1} - \frac{p^{N+1} - 1}{(p - 1)^2}$$

Chương trình tham khảo:

<pre> #include <iostream> #include <math.h> using namespace std; int n, p; long long s; int main() { cin >> n >> p; s = ((n + 1) * (long long) pow(p, n + 1)) / (p - 1) - ((long long) pow(p, n + 1) - 1) / ((p - 1) * (p - 1)); cout << s; return 0; } </pre>	<pre> # Tong day dac biet 3 p = int(input("Nhap p: ")) n = int(input("Nhap n: ")) s = ((n + 1) * p ** (n + 1)) // (p - 1) - (p ** (n + 1) - 1) // ((p - 1) ** 2) print(s) </pre>

PHẦN 2. CẤU TRÚC RỄ NHÁNH

Bài 16. Chẵn lẻ

Cho N. Kiểm tra N là số chẵn hay số lẻ.

❖ **Thuật toán:**

N là số chẵn khi N chia hết cho 2. Trong C++, ta mô tả bằng biểu thức $N \% 2 == 0$.

❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int n; int main() { cin >> n; if (n % 2 == 0) { cout << "n is even"; } else { cout << "n is odd"; } return 0; }</pre>	<pre>n = int(input("Nhập n: ")) if n % 2 == 0: print("N là số chẵn") else: print("N là số lẻ")</pre>

Bài 17. Chia hết cho 3

Cho N. Kiểm tra N có chia hết cho 3 và không chia hết cho 5 hay không? Nếu có ghi ra True ngược lại ghi ra False.

❖ **Thuật toán:**

N chia hết cho 3 biểu diễn là $N \% 3 == 0$

N không chia hết cho 5 biểu diễn là: $N \% 5 != 0$

Phép toán và trong C++: `&&`

Vậy biểu thức điều kiện là $(N \% 3 == 0) \&\& (N \% 5 != 0)$

❖ Chương trình tham khảo:

<pre>include <iostream> using namespace std; int n; int main() { cin >> n; if ((n % 3 == 0) && (n % 5 != 0)) { cout << "True"; } else { cout << "False"; } return 0; }</pre>	<pre>n = int(input("Enter n: ")) if (n % 3 == 0) and (n % 5 != 0): print("True") else: print("False")</pre>

Bài 18. Giá trị lớn nhất

Cho a và b . Tìm giá trị lớn nhất của a và b .

Ví dụ: $a = 5, b = 6$. Kết quả: $rmax = 6$.

❖ Thuật toán:

Ta quan niệm số đầu tiên là giá trị lớn nhất tới thời điểm hiện tại tức là: $rmax = a$.

Sau đó, ta mang giá trị $rmax$ đi so sánh với các giá trị còn lại. Giá trị nào lớn hơn $rmax$ thì cập nhật lại $rmax$ bằng giá trị đó tức là: $if (rmax < b) rmax = b$;

❖ Chương trình tham khảo:

<pre>using namespace std; int a, b;</pre>	<pre>a = int(input("Enter a=")) b = int(input("Enter b=")) rmax = a if rmax < b:</pre>
--	---

<pre> int main() { cin >> a >> b; int rmax = a; if (rmax < b) rmax = b; cout << rmax; return 0; } </pre>	<pre> rmax = b print(rmax) </pre>

Bài 19. Giá trị nhỏ nhất

Cho a, b và c. Tìm giá trị nhỏ nhất của a, b và c.

Ví dụ: a = 5, b = 6, c = 7. Kết quả: rmin = 5.

❖ Thuật toán:

Ta quan niệm số đầu tiên là giá trị nhỏ nhất tới thời điểm hiện tại tức là: $rmin = a$.

Sau đó, ta mang giá trị $rmin$ đi so sánh với các giá trị còn lại. Giá trị nào nhỏ hơn $rmin$ thì cập nhật lại $rmin$ bằng giá trị đó tức là:

$if (rmin > b) rmin = b;$

$if (rmin > c) rmin = c;$

❖ Chương trình tham khảo:

<pre> 1 #include <iostream> 2 3 using namespace std; 4 5 int a, b, c, rmin; 6 7 int main() 8 { 9 cin >> a >> b >> c; 10 rmin = a; 11 if (rmin > b) 12 { 13 rmin = b; 14 } 15 if (rmin > c) 16 { 17 rmin = c; 18 } 19 cout << rmin; 20 return 0; 21 } </pre>	<pre> var a,b,c,rmin: longint; begin readln(a,b,c); rmin := a; if (rmin>b) then rmin := b; if (rmin>c) then rmin := c; write(rmin); end. </pre>
---	---

```

a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))
rmin = a
if rmin > b:
    rmin = b
if rmin > c:
    rmin = c
print(rmin)

```

Bài 20. Cặp số có tổng lớn nhất

Cho a , b và c . Tìm hai số có tổng giá trị lớn nhất.

Ví dụ: $a = 5$, $b = 6$, $c = 3$. Kết quả: $smax = 11$.

❖ Thuật toán:

Ta có các chọn (a, b) , (a, c) và (b, c) . Ta đặt, $m = a + b$; $n = a + c$; $p = b + c$. Bài toán được quy về bài toán tìm giá trị lớn nhất của m , n và p .

Ta quan niệm số đầu tiên là giá trị lớn nhất tới thời điểm hiện tại tức là: $rmax = m$.

Sau đó, ta mang giá trị $rmax$ đi so sánh với các giá trị còn lại. Giá trị nào lớn hơn $rmax$ thì cập nhật lại $rmax$ bằng giá trị đó tức là:

$$\text{if } (rmax < n) \text{ } rmax = n;$$

$$\text{if } (rmax < p) \text{ } rmax = p;$$

Ta đưa $rmax$ là kết quả bài toán ra.

❖ Chương trình tham khảo:

<pre> 1 #include <iostream> 2 3 using namespace std; 4 5 int a, b, c, m, n, p, rmax; 6 7 int main() 8 { 9 cin >> a >> b >> c; 10 m = a + b; 11 p = b + c; 12 n = c + a; 13 rmax = m; 14 if (rmax < n) 15 { 16 rmax = n; 17 } 18 if (rmax < p) 19 { 20 rmax = p; 21 } 22 cout << rmax; 23 return 0; 24 }</pre>	<pre> var a,b,c: longint; smax: longint; begin readln(a,b,c); smax := a+b; if (smax<b+c) then smax := b+c; if (smax<c+a) then smax := c+a; write(smax); end.</pre>
<pre> a = int(input("Enter a: ")) b = int(input("Enter b: ")) c = int(input("Enter c: ")) m = a + b n = b + c p = a + c rmax = m if rmax < n: rmax = n if rmax < p: rmax = p print(rmax)</pre>	

Bài 21. Ghép số

Cho a và b là số nguyên dương có một chữ số. Tìm giá trị lớn nhất của số ghép được từ a và b.

Ví dụ: a = 5, b = 6. Kết quả: res = 65.

❖ Thuật toán:

Ta có các cách ghép (ab) và (ba).

Ở cách ghép 1 chữ số a lên hàng chục, chữ số b hàng đơn vị nên giá trị của cách ghép này là:

$$m = a \cdot 10 + b.$$

Ở cách ghép 2 chữ số b lên hàng chục, chữ số a hàng đơn vị nên giá trị của cách ghép này là:

$$n = b \cdot 10 + a.$$

Bài toán quy về việc tìm giá trị lớn nhất của m và n mà ta đã biết cách giải.

❖ **Chương trình tham khảo:**

<pre> 1 #include <iostream> 2 3 using namespace std; 4 5 int a, b, m, n, rmax; 6 7 int main() 8 { 9 cin >> a >> b; 10 m = a * 10 + b; 11 n = b * 10 + a; 12 rmax = m; 13 if (rmax < n) 14 { 15 rmax = n; 16 } 17 cout << rmax; 18 return 0; 19 }</pre>	<pre> a = int(input("Enter a: ")) b = int(input("Enter b: ")) m = a * 10 + b n = b * 10 + a rmax = m if rmax < n: rmax = n print(rmax)</pre>
<pre> n = int(input("Enter n: ")) a = n // 100 b = (n // 10) % 10 c = n % 10 if a < b: a, b = b, a if b < c: b, c = c, b if a < b: a, b = b, a if b < c: b, c = c, b n = a * 100 + b * 10 + c print(n)</pre>	

Bài 22. Giải phương trình bậc nhất

Cho a và b. Giải và biện luận phương trình $ax + b = 0$.

- Nếu phương trình có Vô số nghiệm thì ghi ra “MULTIPLE”.

- Nếu phương trình Vô nghiệm thì ghi ra “NONE”
- Nếu phương trình có nghiệm thì ghi x ra. Kết quả lấy chính xác 3 chữ số thập phân.

Ví dụ: $a = 5$, $b = 6$. Kết quả: $x = 1.200$

❖ Thuật toán:

Nếu $a = 0$ thì

- Nếu $b = 0$ thì thông báo “MULTIPLE”
- ngược lại, nếu $b \neq 0$ thì thông báo “NONE”;

Ngược lại,

- Tính $x = -\frac{b}{a}$; Sau đó, đưa giá trị của x ra. Lưu ý, x phải là kiểu số thực và viết ra có quy cách bằng câu lệnh `printf("%.3f", x)`;

❖ Chương trình tham khảo:

<pre> 1 #include <iostream> 2 #include <cstdio> 3 4 using namespace std; 5 6 int a, b; 7 float x; 8 9 int main() 10 { 11 cin >> a >> b; 12 if (a == 0) 13 { 14 if (b == 0) 15 { 16 printf("MULTIPLE"); 17 } else 18 { 19 printf("NONE"); 20 } 21 } else 22 { 23 x = -1.0f * b / a; 24 printf("%.3f", x); 25 } 26 return 0; 27 }</pre>	<pre> a = int(input("Enter a: ")) b = int(input("Enter b: ")) if a == 0: if b == 0: print("VSN") else: print("NONE") else: x = -1.0 * b / a print("x = {:.3f}".format(x))</pre>
---	---

Bài 23. Giải phương trình bậc 2

Cho a , b và c . Giải và biện luận phương trình $ax^2 + bx + c = 0$ ($a \neq 0$)

- Nếu phương trình Vô nghiệm thì ghi ra “NONE”.
- Nếu phương trình có Nghiệm kép thì tính và ghi x_{12} ra.
- Nếu phương trình có hai nghiệm phân biệt thì tính và ghi ra x_1 , x_2 ra.

Lấy chính xác đến 2 chữ số thập phân.

Ví dụ: $a = 5$, $b = 6$. Kết quả: $r_{\max} = 6$

❖ **Thuật toán:**

- Ta tính biệt số delta: $d = b^2 - 4ac$;
- Sau đó thực hiện xét dấu của d :
 - Nếu $d < 0$ thì thông báo “NONE”

Ngược lại,

Nếu $d = 0$ thì

{

$$x_{12} = -\frac{b}{2a};$$

Đưa x_{12} ra.

}

Ngược lại thì,

{

$$\text{Tính } x_1 = \frac{-b + \sqrt{d}}{2a}; x_2 = \frac{-b - \sqrt{d}}{2a};$$

Đưa x_1 và x_2 ra.

}

❖ **Chương trình tham khảo:**

```

1  #include <iostream>
2  #include <math.h>
3  #include <cstdio>
4
5  using namespace std;
6
7  int a, b, c;
8  float d, xa, xb;
9
10 int main()
11 {
12     cin >> a >> b >> c;
13     d = 1.0f * b * b - 4.0f * a * c;
14     if (d < 0)
15     {
16         printf("no solution");
17     }
18     else
19         if (d == 0)
20         {
21             printf("x1 = x2 = %0.3f", -1.0 * b / (2.0 * a));
22         }
23     else
24     {
25         xa = (-1.0 * b + sqrt(d)) / (2.0 * a);
26         xb = (-1.0 * b / a) - xa;
27         printf("x1 = %0.3f, x2 = %0.3f", xa, xb);
28     }
29     return 0;
30 }

```

from math import sqrt

a = int(input("Enter a: "))

b = int(input("Enter b: "))

c = int(input("Enter c: "))

d = b * b - 4 * a * c

if d < 0:

 print("NONE")

elif d == 0:

 print("x1 = x2 = {:.3f}".format(-1.0 * b / (2.0 * a)))

else:

 x1 = (-1.0 * b + sqrt(d)) / (2 * a)

 x2 = (-1.0 * b / a) - x1

 print("x1 = {:.3f}".format(x1))

 print("x2 = {:.3f}".format(x2))

Bài 24. Tam giác 1

Cho a, b và c nguyên dương. Kiểm tra a, b, c có là ba cạnh của một tam giác. Nếu có thì tính và đưa ra diện tích tam giác đó, ngược lại ghi ra “NONE”. Kết quả lấy chính xác đến 2 chữ số thập phân.

Ví dụ: a = 3, b = 4, c = 5. Kết quả: S = 6.00

❖ Thuật toán:

- a, b, c là ba cạnh của tam giác khi tổng hai cạnh lớn hơn cạnh còn lại:

$$(a + b > c \ \&\& \ b + c > a \ \&\& \ c + a > b)$$

- Công thức Hê-rông tính diện tích tam giác khi biết độ dài các cạnh:

$$p = \frac{a + b + c}{2}; s = \sqrt{p(p - a)(p - b)(p - c)}$$

- Lưu ý khi tính p ta cần ép kiểu tử số hoặc mẫu số thành kiểu số thực bằng cách nhân với hằng số 1.0

❖ Chương trình tham khảo:

```

1  #include <iostream>
2  #include <cstdio>
3  #include <math.h>
4
5  using namespace std;
6
7  int a, b, c;
8  float p, s;
9
10 int main()
11 {
12     cin >> a >> b >> c;
13     if (a + b <= c || b + c <= a || c + a <= b)
14     {
15         printf("not a triangle");
16     }
17     else
18     {
19         p = (a + b + c) / 2.0;
20         s = sqrt(p * (p - a) * (p - b) * (p - c));
21         printf("%.3f", s);
22     }
23     return 0;
24 }
```

```
from math import sqrt
```

```
a = int(input("Enter a: "))
```

```
b = int(input("Enter b: "))
```

```
c = int(input("Enter c: "))
```

```
if (a + b > c) and (b + c > a) and (c + a > b):
```

```
    p = (a + b + c) / 2.0
```



```
s = sqrt(1.0 * p * (p - a) * (p - b) * (p - c))  
print("Result: {:.3f}".format(s))  
else:  
    print("Not 3 side lengths of a triangle")
```

Bài 25. Điểm và đường tròn

Cho điểm $M(x_1, y_1)$ và đường tròn tâm $I(x_2, y_2)$ bán kính R . Hãy cho biết điểm M nằm trong hay trên hay ngoài đường tròn.

❖ Thuật toán:

- Ta tính khoảng cách từ điểm M tới tâm I của đường tròn:

$$d_{MI} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Ta so sánh khoảng cách này với bán kính của đường tròn. Nếu $d_{MI} > R$ thì M nằm ngoài đường tròn. Nếu $d_{MI} = R$ thì M nằm trên đường tròn. Nếu $d_{MI} < R$ thì M nằm trong đường tròn.

❖ Chương trình tham khảo:

```

1  #include <iostream>
2  #include <math.h>
3
4  using namespace std;
5
6  int m_x, m_y, i_x, i_y, r;
7  float dist_i_m;
8
9  int main()
10 {
11     cin >> m_x >> m_y >> i_x >> i_y >> r;
12     dist_i_m = sqrt((m_x - i_x)*(m_x - i_x) + (m_y - i_y)*(m_y - i_y));
13     if (dist_i_m < r)
14     {
15         cout << "M is inside the circle";
16     }
17     else
18     {
19         if (dist_i_m == r)
20         {
21             cout << "M is on the circle";
22         }
23         else
24         {
25             cout << "M is outside the circle";
26         }
27     }
28     return 0;
29 }
```

```

from math import sqrt
from math import pow
def distance_points(ax, ay, bx, by):
    return sqrt(pow(bx - ax, 2) + pow(by - ay, 2))
mx = int(input("Nhap mx: "))
my = int(input("Nhap my: "))
ix = int(input("Nhap ix: "))
iy = int(input("Nhap iy: "))
```

```

r = int(input("Nhap r: "))
d = distance_points(mx, my, ix, iy)
if d < r:
    print("M nam trong duong tron")
elif d == r:
    print("M nam tren duong tron")
else:
    print("M nam ngoai duong tron")

```

Bài 26. Tam giác 2

Cho điểm $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Hãy kiểm tra A, B, C có là ba đỉnh của tam giác. Nếu có thì tính và đưa ra diện tích tam giác ABC, ngược lại thì ghi ra “NONE”.

❖ Thuật toán:

- Ta tính độ dài các cạnh AB, AC, BC theo công thức

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$d_{AC} = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}$$

$$d_{BC} = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

- Để thuận lợi ta nên tổ chức chương trình con dạng hàm để tính độ dài đoạn thẳng.
- d_{AB} , d_{AC} , d_{BC} là ba cạnh của tam giác khi tổng hai cạnh lớn hơn cạnh còn lại.

❖ Chương trình tham khảo:

```

1  #include <iostream>
2  #include <math.h>
3  #include <cstdio>
4
5  using namespace std;
6
7  int ax, ay, bx, by, cx, cy;
8  float dab, dbc, dca, p, s;
9
10 float distance_points(int mx, int my, int nx, int ny)
11 {
12     return (sqrt((mx - nx) * (mx - nx) + (my - ny) * (my - ny)));
13 }
14
15 int main()
16 {
17     cin >> ax >> ay >> bx >> by >> cx >> cy;
18     dab = distance_points(ax, ay, bx, by);
19     dbc = distance_points(bx, by, cx, cy);
20     dca = distance_points(cx, cy, ax, ay);
21     if (dab + dbc <= dca || dbc + dca <= dab || dca + dab <= dbc)
22     {
23         printf("Not a triangle");
24     }
25     else
26     {
27         p = (dab + dbc + dca) / 2.0;
28         s = sqrt(p * (p - dab) * (p - dbc) * (p - dca));
29         printf("%.3f", s);
30     }
31     return 0;
32 }

```

```
from math import sqrt
```

```
from math import pow
```

```
def distance_points(mx, my, nx, ny):
```

```
    return sqrt(pow(nx - mx, 2) + pow(ny - my, 2))
```

```
ax = int(input("Nhap ax: "))
```

```
ay = int(input("Nhap ay: "))
```

```
bx = int(input("Nhap bx: "))
```

```
by = int(input("Nhap by: "))
```

```
cx = int(input("Nhap cx: "))
```

```
cy = int(input("Nhap cy: "))
```

```
ab = distance_points(ax, ay, bx, by)
```

```
bc = distance_points(bx, by, cx, cy)
```

```
ca = distance_points(cx, cy, ax, ay)
```

```
if (ab + bc > ca) and (bc + ca > ab) and (ca + ab > bc):
```

```
    p = (ab + bc + ca) / 2.0
```

```

s = sqrt(p * (p - ab) * (p - bc) * (p - ca))
print("Ket qua: {:.3f}".format(s))
else:
    print("a,b,c không là ba cạnh của tam giác")

```

Bài 27. Số chẵn lớn nhất

Cho a, b, c và d nguyên dương. Hãy tìm số chẵn lớn nhất. Nếu có thì đưa kết quả tìm được ra, ngược lại ghi ra “NONE”.

Ví dụ: a = 3, b = 2, c = 4, d = 3. Kết quả: res = 4.

❖ Thuật toán:

- Khởi tạo giá trị res ban đầu bằng số vô cùng bé. Vì a, b, c và d nguyên dương nên ta khởi tạo res = 0.
- Lần lượt so sánh res với các số còn lại, nếu số được so sánh là số chẵn và lớn hơn res thì ta cập nhật res.

❖ Chương trình tham khảo:

```

1  #include <iostream>
2  using namespace std;
3  int a, b, c, d, rmax;
4  int main()
5  {
6      cin >> a >> b >> c >> d;
7      rmax = 0;
8      if (a % 2 == 0 && rmax < a)
9      {
10         rmax = a;
11     }
12     if (b % 2 == 0 && rmax < b)
13     {
14         rmax = b;
15     }
16     if (c % 2 == 0 && rmax < c)
17     {
18         rmax = c;
19     }
20     if (d % 2 == 0 && rmax < d)
21     {
22         rmax = d;
23     }
24     if (rmax > 0)
25     {
26         cout << rmax;
27     }
28     else
29     {
30         cout << "value not found";
31     }
32     return 0;
33 }

```

```

var a,b,c,d,res: longint;
begin
    readln(a,b,c,d);
    res := 0;
    if (a mod 2 = 0) and (res<a) then
        res := a;
    if (b mod 2 = 0) and (res<b) then
        res := b;
    if (c mod 2 = 0) and (res<c) then
        res := c;
    write(res);
end.

```

<pre> a = int(input("Enter a: ")) b = int(input("Enter a: ")) c = int(input("Enter a: ")) d = int(input("Enter a: ")) rmax = 0 if (a % 2 == 0) and (rmax < a): rmax = a if (b % 2 == 0) and (rmax < b): rmax = b if (c % 2 == 0) and (rmax < c): rmax = c if (d % 2 == 0) and (rmax < d): rmax = d if rmax > 0: print(rmax) else: print("NONE") </pre>	
--	--

Bài 28. Chữ số lớn nhất

Cho số nguyên dương N là số có 3 chữ số. Hãy tìm chữ số lớn nhất trong N .

Ví dụ: $N = 213 \rightarrow$ Kết quả: $res = 3$.

❖ Thuật toán:

- Ta thực hiện tách các chữ số của N lưu vào a , b và c bằng phép toán chia nguyên và chia dư: $a = N / 100$; $b = (N / 10) \% 10$; $c = N \% 10$;
- Bài toán quy về tìm giá trị lớn nhất của a , b và c (ta đã giải được).

❖ Chương trình tham khảo:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int a, b, c, n, rmax;
6
7  int main()
8  {
9      cin >> n;
10     a = n / 100;
11     b = (n / 10) % 10;
12     c = n % 10;
13     rmax = a;
14     if (rmax < b)
15     {
16         rmax = b;
17     }
18     if (rmax < c)
19     {
20         rmax = c;
21     }
22     cout << rmax;
23     return 0;
24 }
25

```

```

var a,b,c,n: longint;
    m,p,q,rmin: longint;
begin
    readln(n);
    a := n div 100;
    b := (n div 10) mod 10;
    c := n mod 10;
    m := 10*b+c;
    p := 10*a+b;
    q := 10*a+c;
    rmin := m;
    if (rmin>n) then
        rmin := n;
    if (rmin>p) then
        rmin := p;
    if (rmin>q) then
        rmin := q;
    write(rmin);
end.

```

```

n = int(input("Enter n: "))
a = n // 100
b = (n // 10) % 10
c = n % 10
res = a
if res < b:
    res = b
if res < c:
    res = c
print(res)

```

Bài 29. Hoán vị chữ số lớn nhất

Cho N là số nguyên dương có ba chữ số. Người ta hoán vị các chữ số để được số mới. Với $N = 213$ ta có các hoán vị là 123, 231, 312, 321, 132 và hoán vị 321 có giá trị lớn.

Hãy tìm hoán vị chữ số có giá trị lớn nhất.

❖ Thuật toán:

- Ta thực hiện tách các chữ số của N lưu vào a , b và c .
- Nếu $a < b$ thì hoán đổi giá trị a và b .
- Nếu $b < c$ thì hoán đổi giá trị b và c .
- Thực hiện xong hai lần hoán đổi này, giá trị thu được có thể chưa lớn nhất nên ta cần thực hiện lại so sánh trên một lần nữa.
- Nếu $a < b$ thì hoán đổi giá trị a và b .
- Nếu $b < c$ thì hoán đổi giá trị b và c .

❖ Video bài giảng: <https://youtu.be/Cm-fMQ7T9Tc>

❖ Chương trình tham khảo:

<pre> 1 #include <iostream> 2 using namespace std; 3 int a, b, c, n, t; 4 int main() 5 { 6 cin >> n; 7 a = n / 100; 8 b = (n / 10) % 10; 9 c = n % 10; 10 if (a < b) 11 { 12 t = a; 13 a = b; 14 b = t; 15 } 16 if (b < c) 17 { 18 t = b; 19 b = c; 20 c = t; 21 } 22 if (a < b) 23 { 24 t = a; 25 a = b; 26 b = t; 27 } 28 if (b < c) 29 { 30 t = b; 31 b = c; 32 c = t; 33 } 34 n = a * 100 + b * 10 + c; 35 cout << n; 36 return 0; 37 }</pre>	<pre> var a,b,c,n: longint; m,p,q,r,s,t,rmax: longint; begin readln(n); a := n div 100; b := (n div 10) mod 10; c := n mod 10; m := 100*b+10*a+c; p := 100*b+10*c+a; q := 100*a+10*c+b; r := 100*a+10*b+c; s := 100*c+10*a+b; t := 100*c+10*b+a; rmax := m; if (rmax<n) then rmax := n; if (rmax<p) then rmax := p; if (rmax<q) then rmax := q; if (rmax<r) then rmax := r; if (rmax<s) then rmax := s; if (rmax<t) then; rmax := t; write(rmax); end.</pre>
<pre> n = int(input("Enter n: ")) a = n // 100 b = (n // 10) % 10 c = n % 10 if a < b:</pre>	

<pre> a, b = b, a if b < c: b, c = c, b if a < b: a, b = b, a if b < c: b, c = c, b n = a * 100 + b * 10 + c print(n) </pre>	
---	--

Bài 30. Xóa số

Cho N là số nguyên dương có ba chữ số. Người ta thực hiện xóa đi một chữ số của N và giữ nguyên thứ tự các chữ còn lại. Với $N = 123$, thực hiện xóa số như trên ta thu được các số 12, 23, 13 và số 12 là giá trị nhỏ nhất thu được.

Cho N . Hãy tìm giá trị nhỏ nhất thu được theo cách xóa trên.

❖ Thuật toán:

- Ta thực hiện tách các chữ số của N lưu vào a , b và c .
- Với các chữ số thu được ta có thể ghép thành số có hai chữ số theo thứ tự ban đầu là: ab , ac và bc .
- Bài toán qui về tìm giá trị lớn nhất của 3 số nguyên dương.

❖ Chương trình tham khảo:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int n, a, b, c, v1, v2, v3, rmin;
6
7  int main()
8  {
9      cin >> n;
10     a = n / 100;
11     b = (n / 10) % 10;
12     c = n % 10;
13     v1 = a * 10 + b;
14     v2 = b * 10 + c;
15     v3 = a * 10 + c;
16     rmin = v1;
17     if (rmin > v2)
18     {
19         rmin = v2;
20     }
21     if (rmin > v3)
22     {
23         rmin = v3;
24     }
25     cout << rmin;
26     return 0;
27 }

```

```

var a,b,c,n: longint;
    m,p,q,rmin: longint;
begin
    readln(n);
    a := n div 100;
    b := (n div 10) mod 10;
    c := n mod 10;
    m := 10*b+c;
    p := 10*a+b;
    q := 10*a+c;
    rmin := m;
    if (rmin>n) then
        rmin := n;
    if (rmin>p) then
        rmin := p;
    if (rmin>q) then
        rmin := q;
    write(rmin);
end.

```

day3b4

```

n = int(input("Enter n: "))
a = n // 100
b = (n // 10) % 10
c = n % 10
v1 = a * 10 + b
v2 = b * 10 + c
v3 = a * 10 + c
rmin = v1
if rmin > v2:
    rmin = v2
if rmin > v3:
    rmin = v3
print(rmin)

```

Bài 31. Giải phương trình 3

Nhập a, b, c, d, m nguyên dương. Giải và biện luận phương trình: $\frac{ax+b}{cx+d} = m$

- Nếu phương trình Vô số nghiệm ghi ra ‘Multiple’.
- Nếu phương trình Vô nghiệm ghi ra ‘None’.
- Nếu phương trình có nghiệm ghi ra dạng $x = p / q$. Với p, q là các số nguyên.

❖ Thuật toán:

- Ta biến đổi phương trình trên về dạng phương trình bậc nhất một ẩn $p.x + q = 0$.
Với $p = a - e*c$; $q = b - e*d$; và $x \neq -d/c$.
- Nếu $p = 0$ thì
 - Nếu $q = 0$ thì thông báo “Multiple”
 - Ngược lại, thông báo “None”
- Ngược lại,
 - Tính $x = -q/p$.
 - Nếu $x \neq -d/c$ thì cout << -q << “/” << p;

❖ Chương trình tham khảo:

```

1  #include <iostream>
2  using namespace std;
3  int a, b, c, d, e, p, q;
4  float x;
5  int main()
6  {
7      cin >> a >> b >> c >> d >> e;
8      if (d == 0 && e == 0)
9      {
10         cout << "error: invalid input";
11     }
12     else
13     {
14         p = a - e * c;
15         q = b - e * d;
16         if (p == 0)
17         {
18             if (q == 0)
19             {
20                 cout << "infinitely many solutions";
21             }
22             else
23             {
24                 cout << "no solution";
25             }
26         }
27         else
28         {
29             x = -1.0 * p / q;
30             if (x == -1.0 * d / c)
31             {
32                 cout << "no solution";
33             }
34             else
35             {
36                 cout << -p << "/" << q;
37             }
38         }
39     }
40     return 0;
41 }
42

```

```
a = int(input("Enter a: "))
```

```
b = int(input("Enter b: "))
```

```
c = int(input("Enter c: "))
```

```
d = int(input("Enter d: "))
```

```
e = int(input("Enter e: "))
```

```
if (d == 0) and (e == 0):
```

```
    print("NONE")
```

```
else:
```

```
    p = a - e * c
```

```
    q = e * d - b
```

```
    if p == 0:
```

```
        if q == 0:
```

```
            print("MULTIPLE")
```

```

else:
    print("NONE")
else:
    x = -1.0 * p / q
    if x == -1.0 * d / c:
        print("NONE")
    else:
        if p < 0:
            p *= -1
            q *= -1
        print("Result: ", q, "/", p)

```

Bài 32. Điểm trong hình chữ nhật

Cho điểm $M(x_M, y_M)$ và hình chữ nhật có góc trái trên $A(x_1, y_1)$ góc phải dưới $C(x_3, y_3)$.

Kiểm tra M nằm trong hay ngoài hình chữ nhật.

Ví dụ

3 0

0 2

2 0

➔ M nằm trong hình chữ nhật

❖ Thuật toán:

Điểm M nằm trong hình chữ nhật khi $(x_M \geq x_1 \ \&\& \ x_M \leq x_3) \ \&\& \ (y_M \geq y_3 \ \&\& \ y_M \leq y_1)$

❖ Chương trình tham khảo:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int x_M, x1, x3, y_M, ay, y3;
6
7  int main()
8  {
9      cin >> x1 >> ay >> x3 >> y3 >> x_M >> y_M;
10     if (x_M < x1 || x_M > x3 || y_M > ay || y_M < y3)
11     {
12         cout << "M is not in rectangle";
13     }
14     else
15     {
16         cout << "M is in rectangle";
17     }
18     return 0;
19 }

```

```
var xm,ym,x1,y1,x3,y3: longint;
```

```
begin
```

```
    readln(x1,y1,x3,y3,xm,ym);
```

```
    if (x1<=xm) and (xm<=x3) and (y3<=ym) and (ym<=y1) then
```

```
        write('M nam trong hcn')
```

```
    else
```

```
        write('M nam ngoai hcn');
```

```
end.
```

```
x1 = int(input("Enter x1: "))
```

```
y1 = int(input("Enter y1: "))
```

```
x3 = int(input("Enter x3: "))
```

```
y3 = int(input("Enter y3: "))
```

```
xm = int(input("Enter xm: "))
```

```
ym = int(input("Enter ym: "))
```

```
if (xm < x1) or (xm > x3) or (ym < y3) or (ym > y1):
```

```
    print("M is not in rectangle")
```

```
else:
```

```
    print("M is in rectangle")
```

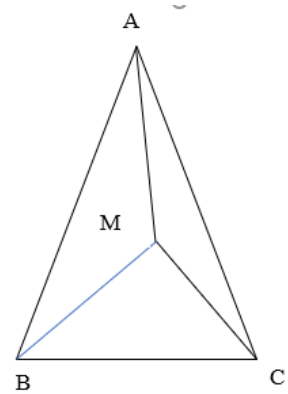
Bài 33. Điểm trong tam giác

Cho tam giác ABC có tọa độ các đỉnh là A(x1, y1), B(x2, y2), C(x3, y3). Xét điểm M(xM, yM) nằm trong hay ngoài tam giác.

Ví dụ: A(0, 0), B(5, 0), C(4, 0), M(1, 1) → Điểm M nằm trong tam giác ABC.

❖ **Thuật toán:**

- Nối MA, MB, MC ta chia tam giác ABC thành 3 tam giác nhỏ MAB, MBC và MCA.
- Điểm M nằm trong tam giác khi tổng diện tích các tam giác này bằng diện tích tam giác ABC.
- Lưu ý, khi tính toán dữ liệu trả về kiểu số thực nên có sai số vì thế biểu thức điều kiện là $S_1 + S_2 + S_3 \leq S$.



❖ **Chương trình tham khảo:**

```
#include <iostream>
#include <math.h>

using namespace std;

float distance_points(int mx, int my, int nx, int ny)
{
    return (sqrt((mx - nx) * (mx - nx) + (my - ny) * (my - ny)));
}

float area_triangle(int a1, int a2, int b1, int b2, int c1, int c2)
{
    float dab = distance_points(a1, a2, b1, b2);
    float dbc = distance_points(b1, b2, c1, c2);
    float dca = distance_points(c1, c2, a1, a2);
    float p = (dab + dbc + dca) / 2.0;
    return sqrt(p * (p - dab) * (p - dbc) * (p - dca));
}

int ax, ay, bx, by, cx, cy, mx, my;
float s, s_1, s_2, s_3;

int main()
{
```



```

cin >> ax >> ay >> bx >> by >> cx >> cy >> mx >> my;

s = area_triangle(ax, ay, bx, by, cx, cy);
s_1 = area_triangle(ax, ay, bx, by, mx, my);
s_2 = area_triangle(mx, my, bx, by, cx, cy);
s_3 = area_triangle(ax, ay, mx, my, cx, cy);

// cout << s_1 + s_2 + s_3;
// cout << s;

if (s_1 + s_2 + s_3 <= s)
{
    cout << "M is in triangle";
}
else
{
    cout << "M is not in triangle";
}

return 0;
}

```

```

from math import sqrt

```

```

def distance_points(mx, my, nx, ny):
    return sqrt((nx - mx) ** 2 + (ny - my) ** 2)

```

```

def area_triangle(ax, ay, bx, by, cx, cy):
    a = distance_points(ax, ay, bx, by)
    b = distance_points(bx, by, cx, cy)
    c = distance_points(ax, ay, cx, cy)
    p = (a + b + c) / 2.0
    return sqrt(1.0 * p * (p - a) * (p - b) * (p - c))

```

```
x1 = int(input("Enter x1: "))
y1 = int(input("Enter y1: "))
x2 = int(input("Enter x2: "))
y2 = int(input("Enter y2: "))
x3 = int(input("Enter x3: "))
y3 = int(input("Enter y3: "))
xm = int(input("Enter xm: "))
ym = int(input("Enter ym: "))
s = area_triangle(x1, y1, x2, y2, x3, y3)
s1 = area_triangle(x1, y1, x2, y2, xm, ym)
s2 = area_triangle(x1, y1, xm, ym, x3, y3)
s3 = area_triangle(xm, ym, x2, y2, x3, y3)
if s1 + s2 + s3 - s <= 1.0e-7:
    print("M is in triangle")
else:
    print("M is not in triangle")
```

PHẦN 3. CẤU TRÚC LẶP

I. Câu lệnh lặp For

Câu lệnh For:

for i in range(1, n):

Các lệnh;

Hoạt động: Cho i chạy từ 1 tới N – 1 và thực hiện <câu lệnh>.

Bài 34. Tổng các phân số liên tiếp

Cho N nguyên dương. Hãy tính $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$. Kết quả lấy chính xác đến 3 chữ số thập phân.

Ví dụ: N = 2, S = 1.500

❖ Thuật toán:

Với i = 1 thì s tăng 1

Với i = 2 thì s tăng $\frac{1}{2}$

Với i = 3 thì s tăng $\frac{1}{3}$

...

Với i thì s tăng $\frac{1}{i}$

Vậy, s += 1/i;

Lưu ý vì 1 và i là số nguyên nên phép chia 1/i là phép chia nguyên. Nên ta phải ép tử số hoặc mẫu số thành số thực bằng cách nhân với hằng số 1.00

❖ Chương trình tham khảo:

```
#include <iostream>
```

```
#include <cstdio>
```

```
using namespace std;
```

```
int n, i;
```

```
float s;
```

```
var i,n: longint;
```

```
s: real;
```

```
begin
```

```
  readln(n);
```

```
  s := 0;
```

```
  for i := 1 to n do
```

```
    s := s+1/i;
```

<pre> int main() { cin >> n; s = 0; for (i = 1; i <= n; i += 1) { s += 1.0 / i; } printf("%0.3f", s); return 0; } </pre>	<pre> write(s:0:3); end. </pre>
<pre> n = int(input("Enter n: ")) s = 0 for i in range(1, n + 1): s += 1.0 / i print("Result: {:.3f}".format(s)) </pre>	

Bài 35. Tổng các phân số lẻ liên tiếp

Cho N nguyên dương. Hãy tính $S = 1 + \frac{1}{3} + \frac{1}{5} \dots + \frac{1}{2N-1}$. Kết quả lấy chính xác đến 3 chữ số thập phân.

Ví dụ: $N = 2, S = 1.833$

❖ Thuật toán:

Với $i = 1$ thì s tăng 1

Với $i = 3$ thì s tăng $\frac{1}{3}$

Với $i = 5$ thì s tăng $\frac{1}{5}$

...

Với i thì s tăng $\frac{1}{i}$ với bước nhảy $i = i + 2$.

Vậy, $s += 1/i; i = i + 2$.

Lưu ý vì 1 và i là số nguyên nên phép chia $1/i$ là phép chia nguyên. Nên ta phải ép tử số hoặc mẫu số thành số thực bằng cách nhân với hằng số 1.00

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <cstdio> using namespace std; int n, i; float s; int main() { cin >> n; s = 0; for (i = 1; i <= n; i += 2) { s += 1.0 / i; } printf("%0.3f", s); return 0; } </pre>	<pre> var i,n: longint; s: real; begin readln(n); s := 0; for i := 1 to n do s := s+1/(2*i-1); write(s:0:2); end. </pre>
<pre> n = int(input("Enter n: ")) s = 0 for i in range(1, n + 1): s += 1.0 / (i * 2 - 1) print("Result: {:.3f}".format(s)) </pre>	

Bài 36. Giai thừa của N

Cho N nguyên dương. Hãy tính $S = 1.2.3.4 \dots N$.

Ví dụ: N = 3, S = 6.

❖ Thuật toán:

S

Với $i = 1$ thì s nhân với 1

Với $i = 2$ thì s nhân với 2

Với $i = 3$ thì s nhân với 3

...

Với i thì s nhân với i .

Vậy, $s *= i$;

Lưu ý, vì S là tích của các số tự nhiên từ 1 tới N nên S có thể rất lớn. Ta cho S kiểu nguyên **long long**.

❖ Chương trình tham khảo:

```
#include <iostream>
#include <cstdio>

using namespace std;

int n, i;
long long s;

int main()
{
    cin >> n;
    s = 1;
    for (i = 1; i <= n; i += 1)
    {
        s *= i;
    }
    printf(s);
    return 0;
}
```

```
var i,n: longint;
    s: real;
begin
    readln(n);
    s := 1;
    for i := 1 to n do
        s := s*i;
    write(s:0:2);
end.
```

```

n = int(input('Enter n: '))
p = 1
for i in range(1, n + 1):
    p *= i
print(p)

```

Bài 37. Chia hết cho 2 và không chia hết cho 3

Cho N nguyên dương. Hãy liệt kê các số nhỏ hơn hoặc bằng N chia hết cho 2 và không chia hết cho 3.

Ví dụ: N = 10, Kết quả: 2 4 8 10

❖ Thuật toán:

Duyệt i từ 1 tới N

Nếu (i chia hết cho 2) và (i không chia hết cho 3) thì

Viết i ra màn hình.

Phép toán và (and) trong C++ là: **&&**

❖ Chương trình tham khảo:

	<pre> var i,n: longint; begin readln(n); for i := 0 to n do if (i mod 2 = 0) and (i mod 3 <> 0) then write(i,#32); end. </pre>
<pre> n = int(input('Enter n: ')) for i in range(1, n + 1): if (i % 2 == 0) and (i % 3 != 0): print(i) </pre>	

Bài 38. Đếm số chính phương

Cho N nguyên dương. Hãy đếm số lượng số chính phương nhỏ hơn hoặc bằng N

Ví dụ: N = 10, Kết quả: 3. Giải thích: Các số chính phương 1, 4, 9.

❖ Thuật toán:

Cách 1: Độ phức tạp của thuật toán $O(N)$ ($N \leq 10^6$).

res = 0;

Duyệt (i, 1, N)

Nếu i là số chính phương thì

res++;

Cách 2: Độ phức tạp của thuật toán là $O(1)$.

....

8: 1, 4

9: 1, 4, 9

10: 1, 4, 9

11: 1, 4, 9

12: 1, 4, 9

....

15: 1, 4, 9

16: 1, 4, 9, 16

17: 4 số

...

24: 4 số

25: 5 số

Gọi kết quả là Res. Ta đã chứng minh được $Res = \text{trunc}(\text{sqrt}(N))$;

cout << res;

❖ Video bài giảng: <https://youtu.be/VbQsxO1kzL4>

❖ Chương trình tham khảo:

	<pre> var i,n: longint; begin readln(n); for i := 1 to n do if (sqrt(i)=trunc(sqrt(i))) then write(i,#32); end. </pre>
<pre> from math import sqrt n = int(input('Enter n: ')) cnt = 0 for i in range(1, n + 1): if sqrt(i) == int(sqrt(i)): </pre>	

<code>cnt += 1</code>	
<code>print('Result: ', cnt)</code>	

Bài 39. Liệt kê ước nguyên dương

Cho N nguyên dương. Hãy liệt kê các ước nguyên dương của N

Ví dụ: N = 10, Kết quả: 1 2 5 10

❖ **Thuật toán:**

❖ **Video bài giảng:** <https://youtu.be/dCH6bNwvmdo>

❖ **Chương trình tham khảo:**

	<pre>var n,i: longint; begin readln(n); for i := 1 to n do if (n mod i = 0) then write(i,#32); end.</pre>
<pre>n = int(input('Enter n: ')) for i in range(1, n // 2 + 1): if n % i == 0: print(i) print(n)</pre>	

Bài 40. Mua hàng

Cho N (đồng). Món quà A có giá a (đồng), món quà B có giá b(đồng). Nếu phải mua hết đúng N (đồng) các món quà A và món quà B thì mỗi loại mua bao nhiêu sản phẩm. Hãy liệt kê các cách mua.

Ví dụ: N = 10, Kết quả: (2, 2); (5, 0)

❖ **Thuật toán:**

- Gọi x là số lượng món quà A được mua, $0 \leq x \leq N/a$
- Gọi y là số lượng món quà B được mua, $0 \leq y \leq N/b$
- Số tiền đã mua x món quà A và y món quà B là: $a.x + b.y$.

Vì ta mua hết N(đồng) nên ta có biểu thức điều kiện: $ax + by = N$

Vậy, ta `for(x, 0, N/a)`

`For(y,0, N/b)`

Nếu $ax + by = N$ thì

Viết (x, y) ra màn hình.

❖ **Chương trình tham khảo:**

	<pre> var x,y,a,b,n:longint; begin readln(n); for x:=0 to n div a do for y:=0 to n div b do if (a*x+b*y=n) then write(x,#32,y); end. </pre>
<pre> n = int(input('Enter n: ')) a = int(input('Enter a: ')) b = int(input('Enter b: ')) p = False for i in range(n // a + 1): for j in range(n // b + 1): if a * i + b * j == n: print(i, j) p = True if not p: print('NONE') </pre>	

Bài 41. Đổi tiền 1

Cho N (đồng). Jame muốn đổi tiền chỉ gồm các tờ tiền mệnh giá 500, 200 và 100.

Hãy liệt kê các cách rút tiền mà Jame có được.

Nếu không các cách đổi tiền nào thì ghi ra “NONE”.

Ví dụ: $N = 1000$. Kết quả $(0,0,10)$, $(0,1,8)$, $(0,2,6)$, $(0,3,4)$, $(0,4,2)$, $(0,5,0)$, $(1,0,5)$, $(1,1,3)$, $(1,2,1)$, $(2,0,0)$

❖ **Thuật toán:**

Thuật toán: Vết cạn

Cách 1: Độ phức tạp của thuật toán $O(N^3/10^7)$

Gọi a là số lượng tờ tiền 500 rút được ($0 \leq a \leq N/500$)

Gọi b là số lượng tờ tiền 200 rút được ($0 \leq b \leq N/200$)

Gọi c là số lượng tờ tiền 100 rút được ($0 \leq c \leq N/100$)

Đổi N đồng theo yêu cầu đề bài: $500*a + 200*b + 100*c = N$

For(a, 0, N/500)

For(b, 0, N/200)

For(c, 0, N/100)

If ($500*a + 200*b + 100*c == N$)

Cout << a << b << c;

Cách 2: Độ phức tạp của thuật toán $O(N^2/10^5)$

Gọi a là số lượng tờ tiền 500 rút được ($0 \leq a \leq N/500$)

Gọi b là số lượng tờ tiền 200 rút được ($0 \leq b \leq N/200$)

Gọi c là số lượng tờ tiền 100 rút được.

Đổi N đồng theo yêu cầu đề bài: Điều kiện $(N - (500*a + 200*b)) \% 100 == 0$

For(a, 0, N/500)

For(b, 0, N/200)

{

p = N - (500*a + 200*b);

If (p % 100 == 0)

cout << a << b << p/100;

}

❖ Video bài giảng: <https://youtu.be/FouRdKHTxCM>

❖ Chương trình tham khảo:

	<pre> var x,y,z,n:longint; begin readln(n); for x:=0 to n div 500 do for y:=0 to n div 200 do for z:=0 to n div 100 do if (500*n+200*y+100*z=n) then write(x,#32,y,#32,z); end. </pre>
--	--

<pre> n = int(input('Enter n: ')) p = False for t500 in range(n // 500 + 1): for t200 in range(n // 200 + 1): for t100 in range(n // 100 + 1): if t500 * 500 + t200 * 200 + t100 * 100 == n: p = True print(t500, t200, t100) if not p: print('NONE') </pre>	
--	--

Bài 42. Đổi tiền 2

Cho N (đồng). Jame muốn đổi tiền chỉ gồm các tờ tiền mệnh giá 500, 200 và 100.

Hãy đưa ra tổng số tờ tiền nhiều nhất và ít nhất mà Jame có thể có được

Ví dụ: N = 1000. Kết quả: 10 2

❖ Thuật toán:

- Gọi x là số lượng tờ tiền 500 rút được, $0 \leq x \leq N/500$
- Gọi y là số lượng tờ tiền 200 rút được, $0 \leq y \leq N/200$
- Gọi z là số lượng tờ tiền 100 rút được, $0 \leq z \leq N/100$
- Số tiền rút được là: $500x + 200y + 100z$

Vì ta cần rút hết N(đồng) nên ta có biểu thức điều kiện: $500x + 200y + 100z = N$

Vậy, ta for(x, 0, N/500)

For(y, 0, N/200)

For(z, 0, N/100)

Nếu $500x + 200y + 100z = N$ thì

Viết (x, y, z) ra màn hình.

❖ Chương trình tham khảo:

	<pre> uses math; var a,b,c,n,rmin,rmax:longint; begin readln(n); rmin:=n; </pre>
--	--

	<pre> rmax:=0; for a:=0 to n div 500 do for b:=0 to n div 200 do for c:=0 to n div 100 do if (500*a+200*b+100*c=n) then begin rmin:=min(rmin,a+b+c); rmax:=max(rmax,a+b+c); end; write(rmin,#32,rmax); end. </pre>
<pre> n = int(input('Enter n: ')) p = False rmin = n rmax = 0 for t500 in range(n // 500 + 1): for t200 in range(n // 200 + 1): for t100 in range(n // 100 + 1): if t500 * 500 + t200 * 200 + t100 * 100 == n: p = True tmp = t100 + t200 + t500 if tmp > rmax: rmax = tmp if tmp < rmin: rmin = tmp if not p: print('NONE') else: print('min: ', rmin) print('max: ', rmax) </pre>	

Bài 43. Đếm số lượng ước

Cho N. Hãy đếm số lượng ước nguyên dương của N.

Ví dụ: N = 10. Kết quả: 4.

❖ Thuật toán:

Ví dụ: Với N = 100 = 1.100 = 2.50 = 4.25 = 5.20 = 10.10

Khi 1 là ước của 100, thì phần còn lại là 100/1 cũng là ước của 100.

Khi 2 là ước của 100, thì phần còn lại là 100/2 cũng là ước của 100.

....

Khi i là ước của N thì phần còn lại N/i cũng là ước của N.

Vậy, for $i \in [1, \lfloor \sqrt{N} \rfloor]$

Nếu N chia hết cho i thì

{

Res++;

Nếu (i != N/i) thì res++;

}

Đưa res ra.

❖ Chương trình tham khảo:

<pre> n = int(input('Enter n: ')) cnt = 0 i = 1 while i * i <= n: if n % i == 0: cnt += 1 if i != n // i: cnt += 1 i += 1 print('Result:', cnt) </pre>	

Bài 44. Ước chẵn

Liệt kê các ước chẵn nguyên dương của N.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre> n = int(input('Enter n: ')) p = False for i in range(2, n // 2): if (i % 2 == 0) and n % i == 0: p = True print(i) if p: print(n) else: print('NONE') </pre>	

Bài 45. Tổng ước

Liệt kê các ước chẵn nguyên dương của N.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre> n = int(input('Enter n: ')) rsum = 0 i = 1 while i * i <= n: if n % i == 0: rsum += i if i != n // i: rsum += n // i i += 1 print('Result:', rsum) </pre>	

Bài 46. Số nguyên tố

Cho số nguyên dương N. Kiểm tra N có là số nguyên tố. Nếu có ghi ra True, ngược lại ghi ra False.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre>def prime(u): if u <= 1: return False i = 2 while i * i <= u: if u % i == 0: return False i += 1 return True n = int(input('Enter n: ')) print(prime(n))</pre>	

Bài 47. Liệt kê số nguyên tố

Cho số nguyên dương a và b ($a \leq b$). Hãy liệt kê các số nguyên tố thuộc đoạn $[a, b]$.

❖ Thuật toán:

❖ Video bài giảng: <https://youtu.be/X8yJUZuXgA0>

❖ Chương trình tham khảo:

<pre>def prime(u): if u <= 1: return False i = 2 while i * i <= u: if u % i == 0:</pre>	

<pre> return False i += 1 return True a = int(input('Enter start value: ')) b = int(input('Enter stop value: ')) for i in range(a, b + 1): if prime(i): print(i) </pre>	
--	--

II. Câu lệnh lặp while

<p>Cấu trúc câu lệnh while</p> <p>while (điều kiện): câu lệnh;</p> <p>Hoạt động: Chừng nào (điều kiện) còn đúng thì lặp lại thực hiện <câu lệnh></p> <p>Lưu ý: trong <câu lệnh> phải có ít nhất một lệnh tác động vào <điều kiện> để sau khi lặp hữu hạn lần, vòng lặp sẽ dừng.</p>

Bài 48. Tính S

Hãy tính giá trị $s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N} + \dots$

Đến khi giá trị của biểu thức cuối cùng $\frac{1}{N} \leq 0.001$

Đưa S ra.

❖ **Thuật toán:**

❖ **Chương trình tham khảo:**

<pre> n = 1 s = 0 while 1 / n > 0.001: s += 1 / n n += 1 print(s) </pre>	
---	--

Bài 49. Ước chung lớn nhất

Cho số nguyên dương a và b . Hãy tìm ước chung lớn nhất của a và b .

❖ Thuật toán:

Áp dụng giải thuật O'clit để tìm ước chung lớn nhất của hai số nguyên dương.

❖ Chương trình tham khảo:

<pre>def gcd(v0, v1): while v1 > 0: r = v0 % v1 v0 = v1 v1 = r return v0 a = int(input('Enter first value: ')) b = int(input('Enter second value: ')) print('Result:', gcd(a, b))</pre>	

Bài 50. Thừa số nguyên tố

Cho số nguyên dương N . Hãy phân tích N thành tích các thừa số nguyên tố.

Ví dụ: $N = 30$. Kết quả: 2 3 5.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre>def phantich(u): i = 2 while i * i <= u: if u % i == 0: print(i, end=' ') u //= i else: i += 1 if u > 1:</pre>	

<pre> print(u, end=' ') n = int(input('Enter a number: ')) print('Result:') phantich(n) </pre>	
--	--

Bài 51. Số nguyên tố 2

Cho số nguyên dương N. Hãy tìm số nguyên tố nhỏ nhất lớn hơn N.

Ví dụ: N = 30. Kết quả: 31.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre> def prime(u): if u <= 1: return False i = 2 while i * i <= u: if u % i == 0: return False i += 1 return True n = int(input('Enter a number: ')) n += 1 while not prime(n): n += 1 print('Result:', n) </pre>	
---	--

Bài 52. Tổng chữ số của N

Cho số nguyên dương N. Hãy tính tổng chữ số của N.

Ví dụ: N = 312. Kết quả: 6.

❖ Thuật toán:

❖ Chương trình tham khảo:

--	--

<pre> n = int(input('Enter a number: ')) s = 0 while n > 0: s += n % 10 n //= 10 print('Result:', s) </pre>	
--	--

Bài 53. Cắt số

Cho số nguyên dương N. Thực hiện lần lượt cắt các chữ số từ phải sang trái của N ta được N mới. Hãy tìm số nguyên tố lớn nhất thu được.

Ví dụ: N = 3122. Kết quả: 31.

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre> def prime(u): if u <= 1: return False i = 2 while i * i <= u: if u % i == 0: return False i += 1 return True n = int(input('Enter a positive integer: ')) // 10 while (n > 0) and not prime(n): n //= 10 print('Result: ', end=") if prime(n): print(n) </pre>	

<pre>else: print('NONE')</pre>	
------------------------------------	--

Bài 54. Ước nguyên tố lớn nhất

Cho số nguyên dương N . Hãy tìm ước nguyên tố lớn nhất của N .

Ví dụ: $N = 10$. Kết quả: 5.

❖ **Thuật toán:**

❖ **Chương trình tham khảo:**

PHẦN 4. KIỂU MẢNG

Bài 55. Mảng một chiều

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy liệt kê các phần tử trong dãy.

Ví dụ: $N = 3$ và dãy 3 1 2. Kết quả: 3 1 2.

❖ **Thuật toán:**

Nhập dãy vào mảng $a[i]$.

Duyệt từ đầu đến cuối dãy, đưa các phần tử ra, mỗi phần tử cách nhau một dấu cách.

❖ **Chương trình tham khảo:**

<pre>#include <bits/stdc++.h> using namespace std; int main() { int n, a[100000]; cin >> n; for (int i = 0; i < n; i++) { cin >> a[i]; cout << a[i] << " "; } return 0; }</pre>	<pre>var i, n: Longint; a : array[-7..100000] of Longint; begin readln(n); // Nhập số lượng for i := 1 to n do //Nhập dãy a_i read(a[i]); for i := 1 to n do write(a[i], #32); end.</pre>
--	---

<pre> n = int(input("Nhap n: ")) a = list() for i in range(1, n + 1): a.append(int(input())) print (a) </pre>	
---	--

Bài 56. Liệt kê phần tử là số chẵn

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy liệt kê phần tử là số chẵn trong dãy.

Ví dụ: $N = 4$ và dãy 3 1 2 5. Kết quả: 1.

❖ Thuật toán:

Nhập dãy vào dãy $a[i]$;

Duyệt từ đầu đến cuối dãy, nếu $a[i]$ là số chẵn thì đưa phần tử đó ra, mỗi phần tử cách nhau một dấu cách.

❖ Chương trình tham khảo:

<pre> #include <bits/stdc++.h> using namespace std; int main() { int n, a[1000000] ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; if (a[i] % 2 == 0) { cout << a[i] << " " ; } } } </pre>	<pre> var i,n: Longint; a : array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); for i := 1 to n Do if a[i] mod 2 = 0 then write(a[i], #32); end. </pre>
--	---

<pre> return 0; } </pre>	
<pre> n = int(input("Nhap n: ")) a = list() for i in range (0, n): a.append(int(input())) for i in range (0, n) : if a[i] % 2 == 0 : print (a[i]) </pre>	

Bài 57. Vị trí phần tử là số chính phương

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Số chính phương là số nguyên dương có giá trị bằng bình phương của một số nguyên dương khác. Các số chính phương đầu tiên là 1, 4, 9, 16...

Hãy đưa ra **vị trí** các phần tử là số chính phương trong dãy.

Ví dụ: $N = 4$ và dãy 3 1 4 5. Kết quả: 2 3.

❖ Thuật toán:

Số chính phương là số có căn bậc hai là một số nguyên.

Biểu thức toán học là: $\sqrt{u} = \lfloor \sqrt{u} \rfloor$.

Mô tả trong Pascal, C++ là: $\text{sqrt}(u) == \text{trunc}(\text{sqrt}(u))$

Nhập dãy vào mảng $a[i]$;

Duyệt từ đầu đến cuối dãy, nếu $a[i]$ là số chính phương thì đưa vị trí i ra, mỗi giá trị tìm được cách nhau một dấu cách.

❖ Chương trình tham khảo:

<pre> #include <bits/stdc++.h> using namespace std; int main() { </pre>	<pre> var n, i : Longint; a : array[-7..1000000] of Longint; begin readln(n); for i := 1 to n do </pre>
---	---

<pre> int n, a[1000000] ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; if (sqrt(a[i]) == trunc(sqrt(a[i]))) { cout << i << " " ; } } return 0; } </pre>	<pre> read(a[i]); for i := 1 to n do if sqrt(a[i]) = trunc(sqrt(a[i])) then write(i,#32); end. </pre>
<pre> from math import sqrt from math import trunc n = int(input("Nhap n: ")) a = list() for i in range (0, n) : a.append(int(input())) for i in range (0, n) : if sqrt(a[i]) == trunc(sqrt(a[i])) : print (i) </pre>	

Bài 58. Đếm phần tử

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy **đếm số lượng** các phần tử trong dãy *chia hết cho 2 nhưng không chia hết cho 3*.

Ví dụ: $N = 4$ và dãy 4 1 6 5. Kết quả: 1.

❖ Thuật toán:

Nhập dãy vào mảng $a[i]$;

$Res = 0$;

Duyệt từ đầu đến cuối dãy, nếu $a[i]$ chia hết cho 2 **và** không chia hết cho 3 thì tăng biến Res lên một đơn vị.

Đưa biến Res ra.

Lưu ý: ở đây ta có biểu thức quan hệ: $(a[i] \% 2 == 0 \ \&\& \ a[i] \% 3 \neq 0)$

Trong Pascal: $(a[i] \bmod 2 = 0) \text{ and } (a[i] \bmod 3 \neq 0)$

❖ **Chương trình tham khảo:**

<pre>#include <bits/stdc++.h> using namespace std; int main() { int n, a[100000], dem = 0 ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; if (a[i] % 2 == 0 && a[i] % 3 != 0) { dem++ ; } } cout << dem; return 0; }</pre>	<pre>var i,n, res : Longint; a : array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); res := 0; for i := 1 to n do if (a[i] mod 2 = 0) and (a[i] mod 3 <> 0) then inc(res); // res := res + 1; write(res); end.</pre>
<pre>from math import sqrt from math import trunc n = int(input("Nhap n: ")) a = list() for i in range (0, n) : a.append(int(input())) for i in range (0, n) : if sqrt(a[i]) == trunc(sqrt(a[i])) : print (i)</pre>	

Bài 59. Tìm phần tử nhỏ nhất

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy tìm phần tử có giá trị nhỏ nhất.

Ví dụ: $N = 4$ và dãy 3 1 2 5. Kết quả: 1.

❖ Thuật toán:

Nhập dãy vào mảng $a[i]$;

Khởi tạo $rmin = trunc(1e9)$;

Duyệt từ đầu đến cuối dãy, nếu $a[i] < rmin$ thì cập nhật lại $rmin = a[i]$;

Đưa $rmin$ ra.

❖ Chương trình tham khảo:

<pre>#include <bits/stdc++.h> using namespace std; int main() { int n, a[100000+7] ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; } int res = trunc(1e9)+7; ; for (int i = 0; i < n; i++) { if (res > a[i]) { res = a[i] ; } } cout << res ; return 0; }</pre>	<pre>var i,n,res : Longint; a :array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); res := a[1]; for i := 1 to n do if res > a[i] then res := a[i]; write(res); end.</pre>
---	--

<pre> n = int(input("Nhap n: ")) a = list() for i in range(0, n): a.append(int(input())) rmin = a[0] for i in range(1, n): rmin = min(rmin, a[i]) print(rmin) </pre>	
--	--

Bài 60. Đếm số lượng phần tử lớn nhất

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy đếm số lượng phần tử có giá trị lớn nhất.

Ví dụ: $N = 6$ và dãy 3 1 4 2 4 3. Kết quả: 2.

❖ Thuật toán:

- Nhập dãy vào mảng $a[i]$;

- *Tìm giá trị lớn nhất:*

Khởi tạo $rmax = -trunc(1e9)$;

Duyệt từ đầu đến cuối dãy, nếu $a[i] > rmax$ thì cập nhật lại $rmax = a[i]$;

- *Đếm số lượng phần tử có giá trị lớn nhất*

$Res = 0$;

Duyệt từ đầu đến cuối dãy, nếu $a[i] = rmax$ thì cập nhật lại $rmax = rmax + 1$;

Đưa $rmax$ ra.

❖ Chương trình tham khảo:

<pre> #include <bits/stdc++.h> using namespace std; int main() { </pre>	<pre> var i,n,res,dem: Longint; a : array[-7..100000] of Longint; begin dem := 0; readln(n); </pre>
---	---

<pre> int n, a[100000 + 7], dem = 0 ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; } int res = a[0] ; for (int i = 0; i < n; i++) { res = max(res, a[i]) ; } for (int i = 0; i < n; i++) { if (a[i] == res) { dem ++ ; } } cout << dem ; return 0; } </pre>	<pre> for i := 1 to n do read(a[i]); res := a[1]; for i := 1 to n do if res < i then res := a[i]; for i := 1 to n do if a[i] = res then inc(dem); write(dem); end. </pre>
<pre> n = int(input("Nhap so luong phan tu: ")) a = list() sum = 0 for i in range(0, n): a.append(int(input())) rmax = a[0] for i in range(1, n): rmax = max(rmax, a[i]) for i in range(0, n): if a[i] == rmax: sum += 1 </pre>	

<code>print(sum)</code>	
-------------------------	--

Bài 61. Tính tổng và trung bình cộng

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy tính tổng và tính trung bình cộng các phần tử của dãy.

Kết quả lấy chính xác đến 3 chữ số thập phân với biến trung bình cộng.

Ví dụ: $N = 6$ và dãy 3 1 4 2 4 3. Kết quả: 17 2.833

❖ Thuật toán:

- Nhập dãy vào mảng $a[i]$;

- **Tính tổng:**

Khởi tạo $resS = 0$;

Duyệt từ đầu đến cuối dãy, cập nhật lại $resS = resS + a[i]$;

- **Tính trung bình cộng:** $resMid = \frac{resS}{N}$

Đưa $resS$ và $resMid$ ra.

Có thể dùng lệnh `printf("%.3f", resMid)`; để đưa biến trung bình cộng kiểu thực ra.

❖ Chương trình tham khảo:

<pre>#include <bits/stdc++.h> using namespace std; int main() { int n, tong = 0 ; int a[100000 + 7] ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; tong += a[i] ; }</pre>	<pre>var i, n, t : Longint; tb : real; a : array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); t := 0; for i := 1 to n do t := t + a[i]; tb := t / n;</pre>
--	---

<pre>double tbc = 1.0*tong / n ; cout << tong; printf(“%0.2f”,tbc); return 0; }</pre>	<pre>write(t,#32 ,tb:0:2); end.</pre>
<pre>n = int(input("Nhap so luong phan tu: ")) a = list() sum = 0 for i in range(0, n): a.append(int(input())) for i in range(0, n): sum += a[i] sum /= n print("{:.3f}" . format(sum))</pre>	

Bài 62. Trung bình cộng 2

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy đếm số lượng phần tử có giá trị lớn hơn hoặc bằng giá trị trung bình cộng của dãy.

Ví dụ: $N = 6$ và dãy 3 1 4 2 4 3. Kết quả: 4

❖ Thuật toán:

- Nhập dãy vào mảng $a[i]$;

- Tính tổng:

Khởi tạo $resS = 0$;

Duyệt từ đầu đến cuối dãy, cập nhật lại $resS = resS + a[i]$;

- Tính trung bình cộng: $resMid = \frac{resS}{N}$

Duyệt từ đầu đến cuối dãy, nếu $a[i] > resMid$ thì đếm $res = res + 1$;

❖ Chương trình tham khảo:

--	--

<pre> #include <bits/stdc++.h> using namespace std; int main() { int n, tong = 0, dem = 0 ; int a[100000 + 7] ; cin >> n ; for (int i = 0; i < n; i++) { cin >> a[i] ; tong += a[i] ; } double tbc = 1.0 * tong / n ; for (int i = 0; i < n; i++) { if (a[i] > tbc) { dem ++ ; } } printf ("%d" , dem) ; return 0; } </pre>	<pre> var i,n,res : Longint; t,tb : real; a : array[-7..100000] of Longint; begin readln(n); t := 0; res := 0; for i := 1 to n do read(a[i]); for i := 1 to n do t := t + a[i]; tb := t / n ; for i := 1 to n do if a[i] >= tb then inc(res); write(res); end. </pre>
<pre> n = int(input("Nhap so luong phan tu: ")) a = list() sum = 0 dem = 0 for i in range(0, n): a.append(int(input())) for i in range(0, n): sum += a[i] </pre>	

<pre> sum /= n for i in range(0, n): if a[i] >= sum : dem += 1 print(dem) </pre>	
---	--

Bài 63. Vị trí K

Cho số nguyên dương N , K và dãy a_1, a_2, \dots, a_N .

Hãy đưa ra vị trí đầu tiên của K trong dãy số. Nếu không có K trong dãy, ghi ra -1.

Ví dụ: $N = 6$, $K = 4$ và dãy 3 1 4 2 4 3. Kết quả: 3

❖ Thuật toán:

- Nhập dãy vào mảng $a[i]$;
- Duyệt từ đầu đến cuối dãy, nếu $a[i] == k$ thì đưa i ra và *break*;
- Sử dụng kỹ thuật cầm cờ để phát hiện trường hợp vô nghiệm.

❖ Chương trình tham khảo:

<pre> #include <bits/stdc++.h> using namespace std; int main() { int n, k, a[100000 + 7], p = 0 ; cin >> n >> k ; for (int i = 0; i < n; i++) { cin >> a[i] ; } for (int i = 0; i < n; i++) { if (a[i] == k) { printf ("%d" , i + 1) ; p = 1 ; </pre>	<pre> var i,n,k,p : Longint; a : array[-7..100000] of Longint; begin readln(n,k); for i := 1 to n do read(a[i]); p := 0; for i := 1 to n do if a[i] = k then begin write(i); inc(p); break; end; </pre>
---	---

<pre> break ; } } if (p == 0) { printf ("-1") ; } return 0; } </pre>	<pre> if p = 0 then write(-1); end. </pre>
<pre> n = int(input("Nhap so luong phan tu: ")) k = int(input("Nhap so K: ")) a = list() for i in range(0, n): a.append(int(input())) p = 0 for i in range(0, n): if a[i] == k: p = 1 print(i + 1) break if p == 0: print(-1) </pre>	

Bài 64. Chênh lệch lớn nhất

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy tìm độ chênh lệch lớn nhất giữa hai phần tử trong dãy trên.

Ví dụ: $N = 6$ và dãy 3 1 4 2 4 3. Kết quả: 3

❖ Thuật toán:

❖ Chương trình tham khảo:

--	--

<pre> #include <bits/stdc++.h> using namespace std; int main() { int n, a[100000 + 7], rmax = trunc(-1e9) - 7, rmin = trunc(1e9) + 7 ; cin >> n ; for (int i = 1; i <= n; i++) { cin >> a[i] ; rmax = max(rmax, a[i]) ; rmin = min(rmin, a[i]) ; } int res = rmax - rmin ; printf ("%d" , res) ; return 0; } </pre>	<pre> var i, n ,rmax,rmin: Longint; a : array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); rmax := a[1]; rmin := a[1]; for i := 1 to n do if rmax < a[i] then rmax := a[i]; for i := 1 to n do if rmin > a[i] then rmin := a[i]; write(abs(rmax-rmin)); end. </pre>
<pre> n = int(input("Nhap so luong phan tu: ")) a = list() for i in range(0, n): a.append(int(input())) rmin = a[0] rmax = a[0] for i in range(1, n): rmin = min(rmin, a[i]) rmax = max(rmax, a[i]) chenhlech = rmax - rmin print(chenhlech) </pre>	

Bài 65. Tổng lớn nhất

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Tìm dãy 3 phần tử *liên tiếp* có tổng lớn nhất. Đưa tổng lớn nhất tìm được ra.

Ví dụ: Cho $N = 6$; dãy 1 3 **4 2 7** 3 \rightarrow tong = 13.

❖ Thuật toán:

- Xét tổng ba phần tử liên tiếp $a[i] + a[i+1] + a[i+2]$.

- $Res := -trunc(1e9)$;

- Duyệt từ 1 tới $N - 2$:

Nếu $a[i] + a[i+1] + a[i+2] > res$ thì

$Res := a[i] + a[i+1] + a[i+2]$;

- Đưa Res ra.

❖ Video bài giảng:

https://youtu.be/NDVc_lzl2Cg

Tóm tắt: https://youtu.be/4_ykjaHrxq0

❖ Chương trình tham khảo:

<pre>#include <iostream> #define NMAX 100000 using namespace std; int arr[NMAX]; int n; int main() { cin >> n; for (int i = 0; i < n; i++) { cin >> arr[i]; } int rsum = arr[0] + arr[1] + arr[2];</pre>	<pre>var j,i,n,res : Longint; a,b :array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); res:= -trunc(1e9); for i := 1 to n do if a[i] + a[i+1] + a[i+2] > res then res :=a[i] + a[i+1] + a[i+2]; writeln(res); end.</pre>
---	---

<pre> for (int i = 1; i <= n - 3; i++) { int cur = arr[i] + arr[i + 1] + arr[i + 2]; if (rsum < cur) rsum = cur; } cout << "Result: " << rsum; return 0; } </pre>	
<pre> n = int(input("Nhap so luong phan tu: ")) a = list() for i in range(0, n): a.append(int(input())) rmax = a[0] for i in range(0, n - 2): sum = 0 sum = a[i] + a[i + 1] + a[i + 2] rmax = max(rmax, sum) print(rmax) </pre>	

Bài 66. Dãy không giảm

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Dãy không giảm là dãy thỏa mãn $a[1] \leq a[2] \leq \dots \leq a[i] \leq \dots \leq a[N-1] \leq a[N]$

Kiểm tra dãy có là dãy không giảm hay không? Nếu có ghi ra 'YES' ngược lại ghi ra 'NO'.

Ví dụ: Cho $N = 6$; dãy 1 3 4 2 7 3 \rightarrow NO

Ví dụ: Cho $N = 6$; dãy 1 3 4 6 7 8 \rightarrow YES

❖ Thuật toán:

- Khởi tạo $P := 1$;
- Duyệt từ 1 tới $N-1$:

Nếu $a[i] > a[i+1]$ thì

Begin

P:= 0; //biến cờ

Break;

End;

- Nếu P = 1 thì

Write(“YES”)

Ngược lại,

Write(“NO”);

❖ **Video bài giảng:**

<https://youtu.be/Fz2xEqNOqWA>

<https://youtu.be/CylJc34PV2A>

<pre> #include <iostream> #define NMAX 100000 using namespace std; int arr[NMAX]; int n; int main() { cin >> n; for (int i = 0; i < n; i++) { cin >> arr[i]; } bool p = true; for (int i = 0; i < n - 1; i++) { </pre>	<pre> var i,n,p : Longint; a : array[-7..100000] of Longint; begin readln(n); for i := 1 to n do read(a[i]); p := 0; for i := 1 to n do if a[i+1] <= a[i] then inc(p) else begin p := 0; break; end; if p = 0 then writeln('no') else write('yes'); end. </pre>

<pre> if (arr[i] > arr[i + 1]) { p = false; break; } } if (p) cout << "YES"; else cout << "NO"; return 0; } </pre>	
<pre> n = int(input("Nhap so luong phan tu: ")) a = list() for i in range(0, n): a.append(int(input())) p = 0 for i in range(0, n - 1): if a[i] > a[i + 1] : p = 1 break if p == 0: print("YES") else: print("NO") </pre>	

Bài 67. Dãy Fibonacci

Dãy số Fibonacci là dãy số được định nghĩa như sau:

$$F[1] = 1; F[2] = 1;$$

$$F[N] = F[N - 1] + F[N - 2] \text{ với } N \geq 3.$$

Cho số nguyên dương N , hãy xác định số $F[N]$.

❖ **Thuật toán:**

- Khởi tạo:

$F[1] := 1;$

$F[2] := 1;$

- Duyệt từ 3 tới N :

$F[i] := F[i-1] + F[i-2];$

- Đưa $F[N]$ ra.

❖ **Video bài giảng:**

<https://youtu.be/jW0FwG9O6As>

<https://youtu.be/X5jgJjFexU>

❖ **Chương trình tham khảo:**

<pre>#include <iostream> #define NMAX 100001 using namespace std; long long arr[NMAX]; int n; int main() { cin >> n; arr[1] = 1; arr[2] = 1; for (int i = 3; i <= n; i++) { arr[i] = arr[i - 1] + arr[i - 2];/*cout << arr[i] << " ";*/ } }</pre>	<pre>var i,n : Longint; f : array[-7..100000] of Longint; begin readln(n); f[1] := 1; f[2] := 1; for i := 3 to n do f[i] := f[i-1] + f[i-2]; write(f[n]); end.</pre>
---	--

<pre> cout << "Result:" << arr[n]; return 0; } </pre>	
<pre> n = int(input("Nhap n: ")) f = list([0] * n) f[0] = 1 f[1] = 1 for i in range(2, n): f[i] = f[i - 1] + f[i - 2] print(f[n - 1]) </pre>	

Bài 68. Dãy Fibonacci nguyên tố

Dãy số Fibonacci là dãy số được định nghĩa như sau:

$$F[1] = 1; F[2] = 1;$$

$$F[N] = F[N - 1] + F[N - 2] \text{ với } N \geq 3.$$

Cho số nguyên dương P ($P \leq 10^{15}$). Hãy xác định số lượng số vừa là số nguyên tố, vừa là số Fibonacci và nhỏ hơn hoặc bằng P .

Bài 69. Dãy đảo

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy đảo ngược dãy số a và đưa dãy đã đảo ra.

Ví dụ: $N = 4$; 1 2 3 4 \rightarrow Kết quả: 4 3 2 1

❖ Thuật toán:

- Thực hiện đổi chỗ $a[i]$ với $a[N-i+1]$ với $i \in [1, N \text{ div } 2]$
- Duyệt từ 1 tới $N \text{ div } 2$

Begin

$T := a[i];$

$A[i] := a[N-i+1];$

$A[N-i+1] := T;$

End;

- Duyệt từ 1 tới N : đưa $a[i]$ ra.

❖ Video bài giảng:

<https://youtu.be/6Cu1yF1UzrM>

❖ Chương trình tham khảo:

Bài 70. Nhân đôi dãy

Cho số nguyên dương N và dãy a_1, a_2, \dots, a_N .

Hãy tạo mảng b có $2xN$ phần tử từ mảng a . Đưa mảng b ra.

Ví dụ: $N = 6$; 1 3 4 2 7 3; \rightarrow Kết quả: 1 3 4 2 7 3 1 3 4 2 7 3

❖ Thuật toán:

- Xây dựng mảng $b[i]$ có $2xN$ phần tử. Trong đó, N phần tử đầu tiên là dãy $a[i]$.

Phần tử từ $[N+1, 2.N]$ cũng là dãy a .

- Duyệt 1 tới N :

$B[i] := a[i];$

-Duyệt từ 1 tới N :

$B[N+i] := a[i];$

- Ta thu được B có $2.N$ phần tử. Đưa mảng B ra:

Duyệt i từ 1 tới $2*N$:

Đưa $b[i]$ ra.

❖ Video bài giảng:

<https://youtu.be/NERvJ3P3ZfQ>

❖ Chương trình tham khảo:

	<pre> var i,n,t: Longint; a : array[1..1000000] of longint; begin readln(n); for i := 1 to n do read(a[i]); for i := 1 to (n div 2) do begin t := a[i]; </pre>

	<pre> a[i] := a[n-i+1]; a[n-i+1] := t; end; for i := 1 to n do write(a[i],#32) end.</pre>
--	---

Bài 71. Mảng hai chiều

1	4	3	4	5
4	5	6	7	8
5	5	5	6	7

Khai báo:

Kiểu dữ liệu Tên mảng[Số lượng hàng][số lượng cột];

Ví dụ:

```
int a[3000][3000];
```

Cho mảng hai chiều kích thước MxN.

Nhập mảng a[i][j].

Đưa mảng vừa nhập ra dạng bảng.

❖ Chương trình tham khảo:

<pre> #include <bits/stdc++.h> using namespace std; int m, n; int a[300][300]; int main() { freopen("bai70.inp", "r", stdin); freopen("bai70.out", "w", stdout); cin >> m >> n; for (int i = 1; i <= m; i++) for (int j = 1; j <= n; j++)</pre>	<pre> var a:array[-3..300,-3..300] of longint; n,m,i,j: longint; begin assign (input,'bai1.inp'); assign (output,'bai1.out'); reset (input); rewrite (output); readln(m,n); for i:=1 to m do for j:=1 to n do read(a[i,j]); for i:=1 to m do begin</pre>
--	--

<pre> cin >> a[i][j]; for (int i = 1; i<=m;i++) { for (int j = 1; j<=n;j++) cout << a[i][j] << " "; cout << endl; } return 0; } </pre>	<pre> for j:=1 to n do write (a[i,j],#32); writeln; end; close(input);close(output); end. </pre>
--	--

Bài 72. Tổng số chẵn

Cho mảng hai chiều a có kích thước MxN.

Tính tổng các phần tử là số chẵn trong mảng. Đưa tổng tính được ra.

❖ Thuật toán:

Duyệt tất cả các phần tử trong mảng, nếu $a[i][j]$ là số chẵn thì ta tăng tổng res lên một lượng là $a[i][j]$.

Trước đó, ta khởi tạo $res = 0$. Lưu ý, kiểu dữ liệu của **res** là **long long**.

❖ Video bài giảng:

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); cin >> m >> n; </pre>	<pre> var i,j,n,m: longint; res : Int64; a : array[-3..300,-3..300] of longint; begin assign(input, 'no2.inp');reset(input); assign(output,'no2.out');rewrite(output); read(m,n); for i := 1 to m do for j := 1 to n do </pre>
---	---

<pre> for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; long long rsum = 0; for (int i = 0; i < m; i++) { for (int j = 0; j < n; j++) if (a[i][j] % 2 == 0) rsum += a[i][j]; } cout << rsum; return 0; } </pre>	<pre> read(a[i,j]); res := 0; for i := 1 to m do begin for j := 1 to n do if a[i,j] mod 2 = 0 then res := res + a[i,j]; end; write(res); close(input);close(output) end. </pre>
---	--

Bài 73. Đếm số lượng

Cho mảng hai chiều a có kích thước MxN và số nguyên K.

Đếm số lượng các phần tử có giá trị bằng K. Đưa kết quả tính được ra.

❖ **Thuật toán:**

Duyệt qua tất các phần tử của mảng, nếu phần tử đó có giá trị bằng K thì tăng biến đếm lên 1 đơn vị.

❖ **Video bài giảng:**

❖ **Chương trình tham khảo:**

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int k; int main() { freopen("data.inp", "r", stdin); </pre>	<pre> var i,j,n,m,k,dem: longint; a : array[-3..300,-3..300] of longint; begin assign(input, 'no3.inp'); reset(input); assign(output, 'no3.out'); rewrite(output); read(m,n,k); for i := 1 to m do </pre>
---	--

<pre> freopen("data.out", "w", stdout); cin >> m >> n >> k; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; int rcnt = 0; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) if (a[i][j] == k) rcnt++; cout << rcnt; return 0; } </pre>	<pre> for j := 1 to n do read(a[i,j]); dem := 0; for i := 1 to m do begin for j := 1 to n do if a[i,j] = k then inc(dem); end ; writeln(dem); end ; close(input);close(output) end . </pre>
--	---

Bài 74. Vị trí phần tử chia hết cho K

Cho mảng hai chiều a có kích thước MxN và số nguyên K.

Đưa ra vị trí những phần tử chia hết cho K.

❖ Thuật toán:

Bài toán chỉ quan tâm $a[i][j]$ có chia hết cho K hay không? Do đó, ta không nhất thiết phải lưu giá trị thực sự của $a[i][j]$ mà chỉ cần lưu phần dư $a[i][j] \% K$.

Khi đó, việc đưa ra các phần tử $a[i][j]$ có chia hết cho K sẽ quy về bài toán: đưa ra vị trí của các phần tử $a[i][j] = 0$.

Duyệt qua tất các phần tử của mảng,

nếu $a[i][j] = 0$ thì đưa chỉ số (i, j) ra.

❖ Video bài giảng:

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; </pre>	<pre> var i,j,m,n,k,x: longint; a : array[-3..300,-3..300] of longint; begin assign(input, 'no4.inp');reset(input); assign(output,'no4.out');rewrite(output); </pre>
--	--

<pre> int main() { freopen("data.inp","r",stdin); freopen("data.out","w",stdout); cin >> m >> n; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; int k; cin >> k; for (int i = 0; i < m; i++) { for (int j = 0; j < n; j++) if (a[i][j] % k == 0) cout << "(" << i + 1 << ", " << j + 1 << ")", "; cout << endl; } return 0; } </pre>	<pre> readln(m,n,k); for i := 1 to m do for j := 1 to n do begin read(x); a[i,j] := x mod k; end; for i := 1 to m do for j := 1 to n do if a[i,j] = 0 then writeln(i,#32,j); close(input);close(output); end. </pre>
<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp","r",stdin); freopen("data.out","w",stdout); int k, x; cin >> m >> n >> k; for (int i = 0; i < m; i++) </pre>	

<pre> for (int j = 0; j < n; j++) { cin >> x; a[i][j] = x%k; } for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) if (a[i][j] == 0) { cout << i + 1 << " " << j + 1 << endl; cout << endl; } return 0; } </pre>	
---	--

Bài 75. Tìm giá trị lớn nhất

Cho mảng hai chiều a có kích thước MxN.

Tìm giá trị lớn nhất của mảng. Đưa giá trị lớn nhất tìm được ra.

❖ Thuật toán:

Khởi tạo, $res = a[1][1]$; // $res = -trunc(1e9)$;

Duyệt qua tất các phần tử,

Nếu $res < a[i][j]$ thì

$res = a[i][j]$;

Đưa res ra.

❖ Video bài giảng:

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { </pre>	<pre> var i,j,m,n,res: longint; a : array[-3..300,-3..300] of longint; begin assign(input, 'no5.inp');reset(input); assign(output, </pre>
---	---

<pre> freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); cin >> m >> n; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; int rmax = a[0][0]; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) if (a[i][j] > rmax) rmax = a[i][j]; cout << rmax; return 0; } </pre>	<pre> 'no5.out');rewrite(output); readln(m,n); for i := 1 to m do for j := 1 to n do read(a[i,j]); res := a[1,1]; for i := 1 to m do for j := 1 to n do if a[i,j] > res then res := a[i,j]; writeln(res); close(output);close(input); end. </pre>
---	---

Bài 76. Đếm số lượng giá trị nhỏ nhất

Cho mảng hai chiều a có kích thước $M \times N$.

Đếm số lượng phần tử đạt giá trị nhỏ nhất của mảng. Đưa giá trị nhỏ nhất và số lượng phần tử đạt giá trị nhỏ nhất ra.

❖ Thuật toán:

- Tìm giá trị nhỏ nhất của mảng:

```
Res = a[1][1]; // res = trunc(1e9);
```

Duyệt qua tất cả các phần tử trong mảng

Nếu $a[i][j] < res$ thì

```
Res = a[i][j];
```

- Đếm số lượng phần tử đạt giá trị nhỏ nhất:

```
dres = 0;
```

Duyệt qua tất cả các phần tử

Nếu $a[i][j] = res$ thì

```
dres++;
```

Đưa res và $dres$ ra.

❖ Video bài giảng:

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); cin >> m >> n; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; int rmin = a[0][0]; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) if (a[i][j] < rmin) rmin = a[i][j]; int rcnt = 0; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) if (a[i][j] == rmin) rcnt++; cout << rmin << " " << rcnt; return 0; } </pre>	
---	--

Bài 77. Tìm giá trị lớn nhất của mỗi hàng

Cho mảng hai chiều a có kích thước MxN.

Tìm giá trị lớn nhất của mỗi hàng. Đưa giá trị lớn nhất của mỗi hàng ra.

3	2	5	6	2
5	6	7	1	1
2	2	2	2	2

Kết quả: 6 7 2

❖ **Thuật toán:**

Duyệt qua tất cả các hàng,

{

rmax = - trunc(1e9); //Tìm giá trị lớn nhất của từng hàng

Với mỗi hàng, ta duyệt qua tất cả các cột

Nếu $a[i][j] > rmax$ thì

rmax = $a[i][j]$;

Đưa rmax;

}

❖ **Video bài giảng:** <https://youtu.be/ybP4FIT1png>

❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); cin >> m >> n; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; for (int i = 0; i < m; i++) { int rmax = a[i][0]; for (int j = 1; j < n; j++) if (a[i][j] > rmax)</pre>	

<pre> rmax = a[i][j]; cout << rmax << endl; } return 0; } </pre>	
--	--

Bài 78. Tìm giá trị nhỏ nhất của mỗi cột

Cho mảng hai chiều a có kích thước MxN.

Tìm giá trị nhỏ nhất của mỗi cột. Đưa giá trị nhỏ nhất của mỗi cột ra.

3	2	5	6	2
5	6	7	1	1
2	2	2	2	2

❖ Thuật toán:

Duyệt qua tất cả các cột,

```

{
    res = trunc(1e9);
    Duyệt qua tất cả các hàng
        Nếu a[i][j] < res thì
            res = a[i][j];
    Đưa res ra;
}

```

❖ Video bài giảng:

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); </pre>	
--	--

```

cin >> m >> n;
for (int i = 0; i < m; i++)
    for (int j = 0; j < n; j++)
        cin >> a[i][j];
for (int i = 0; i < n; i++)
{
    int rmin = a[0][i];
    for (int j = 1; j < m; j++)
        if (a[j][i] < rmin)
            rmin = a[j][i];
    cout << rmin << endl;
}
return 0;
}

```

Bài 79. Tìm tổng từng hàng, từng cột

Cho mảng hai chiều a có kích thước MxN.

Tìm tổng của từng hàng, tổng của từng cột. Đưa các giá trị tính được ra.

3	2	5	6	2
5	6	7	1	1
2	2	2	2	2

Kết quả:

18 20 10

10 10 14 9 5

❖ Thuật toán:

- Tính tổng từng hàng:

Duyệt qua tất cả các hàng,

{

Res = 0;

Duyệt qua tất cả các cột,

Res += a[i][j];

Đưa Res ra;

}

- Tính tổng từng cột:

Duyệt qua tất cả các cột,

```
{
    Res = 0;
    Duyệt qua tất cả các hàng,
        Res += a[i][j];
    Đưa Res ra;
}
```

❖ **Video bài giảng:**

❖ **Chương trình tham khảo:**

<pre>#include <iostream> using namespace std; int a[3000][3000]; int m, n; int main() { freopen("data.inp", "r", stdin); freopen("data.out", "w", stdout); cin >> m >> n; for (int i = 0; i < m; i++) for (int j = 0; j < n; j++) cin >> a[i][j]; for (int i = 0; i < m; i++) { longint rsum = 0; for (int j = 0; j < n; j++) rsum += 1ll*a[i][j]; cout << rsum << endl; } for (int i = 0; i < n; i++) { long long rsum = 0; for (int j = 0; j < m; j++)</pre>	

<pre> rsum += 1ll*a[j][i]; cout << rsum << endl; } return 0; } </pre>	
---	--

Bài 80. Vị trí các số chính phương

Cho mảng hai chiều a có kích thước MxN.

Tìm vị trí các phần tử là số chính phương. Đưa các vị trí tìm được ra.

Nếu không có số chính phương nào trong mảng thì ghi ra -1.

❖ **Thuật toán:**

❖ **Video bài giảng:**

❖ **Chương trình tham khảo:**

- Tổ chức hàm chinhphuong(u) để kiểm tra u là số chính phương hay không?
- Sử dụng kỹ thuật cài cò để phát hiện trường hợp vô nghiệm.
- Khởi tạo P = 0;
- Duyệt qua tất cả các phần tử trong mảng,

Nếu chinhphuong(u) thì

{

Đưa chỉ số i, j ra.

P = 1;

}

- Nếu P = 0 thì ghi ra -1.

<pre> #include <iostream> #include <math.h> using namespace std; int a[3000][3000]; int m, n, p = 0; int main() </pre>	
---	--

```

{
    freopen("data.inp", "r", stdin);
    freopen("data.out", "w", stdout);
    cin >> m >> n;
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            cin >> a[i][j];
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            int v = a[i][j];
            if (sqrt(v) == (int) sqrt(v))
            {
                cout << i + 1 << ", " << j + 1 << endl;
                p = 1;
            }
        }
    }
    if (p == 0)
        cout << -1;
    return 0;
}

```

Bài 81. Vị trí các số nguyên tố

Cho mảng hai chiều a có kích thước $M \times N$.

Tìm vị trí các phần tử là số nguyên tố. Đưa các vị trí tìm được ra.

Nếu không có số nguyên tố nào trong mảng thì ghi ra -1.

❖ **Thuật toán:**

❖ **Video bài giảng:**

❖ **Chương trình tham khảo:**

PHẦN 5. KIỂU XÂU

Xâu là tập hợp các phần tử có cùng kiểu dữ liệu là char.

Sử dụng mảng *char* hoặc chuỗi *string*.

Bài 82. Xâu 1

Nhập chuỗi S. Đưa ra chuỗi S vừa nhập và độ dài của chuỗi S.

Ví dụ: S = "abcde";

Kết quả: abcde 5

❖ Thuật toán:

- Để nhập chuỗi trong Pascal ta dùng lệnh Readln(s);
- Để nhập chuỗi trong C++ ta dùng lệnh: Nếu chuỗi không chứa dấu cách: cin >> s; Nếu chuỗi có dấu cách thì dùng lệnh getline(cin, s);

❖ Chương trình tham khảo:

Code C++	
<pre>#include <iostream> #include <string> using namespace std; int main() { string s; getline(cin, s); cout << s << endl; cout << s.size(); return 0; }</pre>	<pre>var s: string; p: longint; begin readln(s); p := length(s); write(s,#32,p); end.</pre>
<pre>s = str(input("nhap xau s: ")) print(s) print(len(s))</pre>	

Bài 83. Vị trí S1 trong S

Cho chuỗi S1, S. Đưa ra vị trí đầu tiên của chuỗi S1 trong chuỗi S.

Ví dụ: S1 = 'ab', S = 'cbabc'. Kết quả: 3

❖ Thuật toán:

- Trong Pascal ta dùng hàm pos(u, v) để trả về vị trí đầu tiên của chuỗi u trong chuỗi v.

- Trong C++ ta dùng hàm $v.find(u) + 1$ là vị trí của xâu u trong xâu v.

❖ **Chương trình tham khảo:**

<pre>#include <iostream> #include <string> using namespace std; int main() { string s, s1; cout << "Enter first string: "; getline(cin, s); cout << "Enter second string: "; getline(cin, s1); unsigned int rpos = s.find(s1); cout << "Result: " << rpos; return 0; }</pre>	<pre>var a,b : String; begin assign(input, 'no2.inp'); reset(input); assign(output, 'no2.out'); rewrite(output); readln(a); readln(b); write(pos(a, b)); close(input); close(output); end.</pre>
<pre>s1 = str(input("Nhap xau S1: ")) s = str(input("Nhap xau S: ")) print(s.find(s1))</pre>	

Bài 84. Đếm xâu S1 trong xâu S

Cho xâu S1 và S. Đếm số lần xuất hiện xâu S1 trong xâu S.

Ví dụ: S1 = 'ab', S = 'cbabcbab'. Kết quả: 2

❖ **Thuật toán:**

- Trong Pascal ta dùng hàm $pos(u, v)$ để trả về vị trí đầu tiên của xâu u trong xâu v.
- Trong C++ ta dùng hàm $v.find(u) + 1$ là vị trí của xâu u trong xâu v.
- Khởi tạo Res = 0.
- Chừng nào s1 còn xuất hiện trong s2 ta thực hiện:
 - Tăng Res lên 1 đơn vị;
 - Xoá từ đầu xâu S tới vị trí vừa tìm thấy S1 trong S đi 1 kí tự.
- Đưa Res ra.

❖ **Chương trình tham khảo:**

<pre> #include <bits/stdc++.h> #define For(i,a,b) for(int i=a;i<=b;i++) using namespace std; int main() { freopen("bai3.inp","r",stdin); freopen("bai3.out","w",stdout); string s1,s2; cin >> s1 >> s2; int d=0; while(s2.find(s1)+1>0) { d++; s2.erase(1,s2.find(s1)+s1.size()-1); } cout << d; return 0; } </pre>	<pre> var s,s2: String; res : Longint; begin assign(input, 'no3.inp');reset(input); assign(output, 'no3.out'); reset(output); readln(s); readln(s2); res := 0; while pos(s,s2) > 0 do begin inc(res); delete(s2, 1, pos(s,s2)+ length(s)-1); end; close(input);close(output); end. </pre>
<pre> #include <iostream> #include <string> using namespace std; int main() { string s, s1; cout << "Enter string s1: "; getline(cin, s1); cout << "Enter string s: "; getline(cin, s); int counter = 0; </pre>	<pre> var s,s1 : String; function dls (a,b : String): Longint; var res ,p: longint; begin res := 0; while pos(b, a) <> 0 do begin p := pos(b,a); inc(res); delete(a, 1, p); end; </pre>

<pre> size_t idx = s.find(s1); while (idx != string::npos) { counter++; idx = s.find(s1, idx + s1.size()); } cout << "Result: " << counter; return 0; } </pre>	<pre> exit(res); end; begin assign(input, 'no3.inp'); reset(input); assign(output, 'no3.out'); rewrite(output); readln(s); readln(s1); dls(s1, s); close(input); close(output); end. </pre>
--	--

Bài 85. Nối chuỗi

Nhập chuỗi S1, S2. Hãy tạo chuỗi S bằng cách nối S2 vào sau chuỗi S1. Đưa S ra.

Ví dụ: S1 = "abcbabc"; S2 = "bc" → Kết quả: S = "abcbabcbc";

❖ Thuật toán:

- Ta sử dụng phép toán nối chuỗi: phép cộng chuỗi. Lưu ý: Phép cộng chuỗi không có tính chất giao hoán tức là $u + v$ cho kết quả khác $v + u$.

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> using namespace std; int main() { string s, s1, s2; cout << "Enter first string: "; cin >> s1; cout << "Enter second string: "; cin >> s2; s = s1 + s2; </pre>	<pre> var s,s1,s2:string; begin assign(input,'x2.inp'); assign(output,'x2.out'); reset(input); rewrite(output); readln(s1); readln(s2); s:=s1+s2; close(input); close(output); end. </pre>
--	--

<pre>cout << "Result: " << s; return 0; }</pre>	
<pre>s1 = str(input("Nhap s1: ")) s2 = str(input("Nhap s2: ")) s = s1 + s2 print(s)</pre>	

Bài 86. Đếm kí tự số

Nhập vào S chỉ bao gồm các kí tự từ 'a' đến 'z', từ '0' đến '9'. Đếm số lượng kí tự là số trong S.

Ví dụ: S = "ab44cb3a2bc" → Kết quả: 4

❖ Thuật toán:

- Duyệt từ đầu đến cuối chuỗi,
Nếu s[i] là kí tự số thì tăng biến đếm lên.
- Đưa biến đếm ra.

❖ Chương trình tham khảo:

<pre>int main() { string s; string s1 ("0123456789"); cout << "Enter a string: "; getline(cin, s); int counter = 0; size_t pos = s.find_first_of(s1); while (pos != string::npos) { counter++; pos = s.find_first_of(s1, pos + 1); } cout << "Result: " << counter;</pre>	<pre>const fi = 'xau5.inp'; fo = 'xau5.out'; var s: string; i,d : longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); readln(s); d:=0; for i:=1 to length(s) do if (s[i] >= '0') and (s[i] <= '9') then inc(d); write(d);</pre>
---	--

<pre> return 0; } </pre>	<pre> close(input);close(output); end. </pre>
<pre> #include <iostream> using namespace std; int main() { string s ; getline (cin , s) ; int dem = 0 ; for (int i = 0; i < s.size(); i++) { if (s[i] >= '0' && s[i] <= '9'){ dem ++ ; } } cout << dem ; return 0; } </pre>	<pre> def demsokitu (a) : dem = 0 for i in range(0, len(a)): if a[i] >= '0' and a[i] <= '9' : dem += 1 return dem s = str(input("Nhap xau s: ")) print(demsokitu(s)) </pre>

Bài 87. Xâu đảo

Tìm xâu đảo của xâu S.

Ví dụ: S = “abc” → Kết quả: S = “cba”;

❖ Thuật toán:

- Duyệt từ đầu đến giữa xâu,
Đổi chỗ (s[i], s[n-i+1]); với n là độ dài của xâu.
- Đưa xâu S ra.

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> #include <cstring> </pre>	<pre> var s: String; function island (u: String) : String ; var </pre>
---	---

<pre> using namespace std; int main() { string s; cout << "Enter string: "; getline(cin, s); int n = s.size(); char * cs = new char[n + 1]; strcpy(cs, s.c_str()); for (int i = 0; i < n / 2; i++) { char t = cs[i]; cs[i] = cs[n - 1 - i]; cs[n - 1 - i] = t; } s = cs; cout << "Result: " << s; delete[] cs; return 0; } </pre>	<pre> tam : String= ""; i : Longint; begin for i := 1 to length(u) do tam := s[i]+ tam; end; begin assign(input, 'no5.inp'); reset(input); assign(output, 'no5.out'); rewrite(output); readln(s); writeln(island(s)); close(input); close(output); end. </pre>
<pre> #include <bits/stdc++.h> #define For(i,a,b) for(int i=a;i<=b;i++) using namespace std; string sdao(string u) { string tam=""; For(i,0,u.size()-1) tam=u[i]+tam; return tam; } int main() </pre>	<pre> def xaudao(a) : b = list(reversed(a)) c = ".join(b) return c s = str(input("Nhap xau s: ")) print("Dao nguoc cua xau s la: ", xaudao(s)) </pre>

<pre> { freopen("bai6.inp", "r", stdin); freopen("bai6.out", "w", stdout); string s1, s2, s; cin >> s; //std::reverse(s.begin(), s.end()); cout << sdao(s); return 0; } </pre>	
--	--

Bài 88. Xâu đối xứng

Xâu đối xứng (xâu palidrom) là xâu đọc từ trái sang phải hay từ phải sang trái ta được xâu giống nhau. Các xâu đối xứng: a, aba, abba, abbcbbba,....

Cho xâu S. Kiểm tra S có là xâu đối xứng hay không? Nếu có ghi True, ngược lại ghi ra False.

Ví dụ: S = "abc" → Kết quả: False

Ví dụ: S = "aba" → Kết quả: True

❖ Thuật toán:

- Tổ chức hàm tìm xâu đảo của xâu u.
- Một xâu là xâu đối xứng khi nó có giá trị bằng xâu đảo của chính nó: $u == \text{xaudao}(u)$

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> #include <cstring> using namespace std; string str_reverse(string s) { int n = s.size(); string s1; s1 = ""; </pre>	<pre> var s: string; function xaudao(u:string): string; var tam: string=""; i: longint; begin for i := 1 to length(u) do tam := u[i]+tam; exit(tam); end; begin readln(s); </pre>
---	---

<pre> for (int i = 0; i < n; i++) { s1 = s[i] + s1; } return s1; } bool palin(string s) { return s == str_reverse(s); } int main() { string s; cout << "Enter a string: "; cin >> s; if (palin(s)) cout << "True"; else cout << "False"; return 0; } </pre>	<pre> if (s = xaudao(s)) then write(true) else write(false); end. </pre>
<pre> def xaudioixung(a) : d = a b = list(reversed(a)) c = ".join(b) return d == c s = str(input("Nhap xau s: ")) print(xaudioixung(s)) </pre>	

Bài 89. Xâu chữ hoa

Đổi xâu S thành toàn xâu chữ hoa.

Ví dụ: S = “abc” → Kết quả: S = “ABC”;

❖ Thuật toán:

- Để đổi kí tự thường thành kí tự hoa trong Pascal ta sử dụng hàm **upcase**

- Để đổi kí tự thường thành kí tự hoa trong C++ ta sử dụng hàm **toupper**

❖ **Chương trình tham khảo:**

<pre>#include <iostream> #include <string> using namespace std; int main() { string s; cout << "Enter a string: "; getline(cin, s); for (unsigned int i = 0; i < s.size(); i++) s[i] = toupper(s[i]); cout << "Result: " << s; return 0; }</pre>	<pre>var s:string; i:longint; begin assign(input,'x8.inp'); assign(output,'x8.out'); reset(input); rewrite(output); readln(s); for i:=1 to length(s) do s[i]:=uppercase(s[i]); write(s); close(input); close(output); end.</pre>
---	--

Bài 90. Xâu chữ thường

Đổi xâu S thành toàn xâu chữ thường.

Ví dụ: S = “ABC” → Kết quả: S = “abc”;

❖ **Thuật toán:**

- Để đổi kí tự thường thành kí tự thường trong Pascal ta sử dụng hàm *chr(ord(s[i]) + 32)*

- Để đổi kí tự thường thành kí tự thường trong C++ ta sử dụng hàm *tolower(s[i])*;

❖ **Chương trình tham khảo:**

<pre>#include <iostream> #include <string> using namespace std;</pre>	
--	--

<pre> int main() { string s; cout << "Enter a string: "; getline(cin, s); for (unsigned int i = 0; i < s.size(); i++) s[i] = tolower(s[i]); cout << "Result: " << s; return 0; } </pre>	
--	--

Bài 91. Chuẩn hóa xâu

Xóa các dấu cách thừa trong xâu, nếu có nhiều dấu cách đứng cạnh nhau thì giữ lại 1 dấu cách.

Ví dụ: S = “ AB CC DD AA ABC ” → Kết quả: S = “AB CC DD AA ABC”;

❖ Thuật toán:

Ta nối thêm một dấu cách vào đầu xâu, một dấu cách vào cuối xâu.

Đặt xâu s1 = ‘ ‘; //Hai dấu cách.

Chừng nào s1 còn xuất hiện trong S thì

Xoá trong xâu S, từ vị trí tìm thấy s1 đầu tiên, đi 1 kí tự.

Nếu S[1] là dấu cách thì xoá S[1] đi;

Nếu S[n] là dấu cách thì xoá S[n] đi với n là độ dài của xâu.

Đưa xâu S ra.

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> using namespace std; string chuanhoa(string s) { string s2 = s; </pre>	<pre> const fi = 'xau10c2.inp'; fo = 'xau10c2.out'; var s,s1: string; p : longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); </pre>

<pre> string s1; s1 = " "; size_t pos = s2.find(s1); while (pos != string::npos) { s2.erase(pos, 1); pos = s2.find(s1, pos); } if (s2.size() != 0 && s2[0] == ' ') s2.erase(0, 1); if (s2.size() != 0 && s2.back() == ' ') s2.pop_back(); return s2; } int main() { string s; cout << "Enter string: "; getline(cin, s); cout << "Result: " << chuanhoa(s); return 0; } </pre>	<pre> readln(s); s:=' ' + s; s1:= ' '; while pos(s1,s) > 0 do begin p:= pos(s1,s); delete(s,p,1); end; if s[1] = #32 then delete(s,1,1); if s[length(s)] = #32 then delete(s,length(s),10); write(s); close(input);close(output); end. </pre>
--	--

Bài 92. Đếm số lượng từ trong chuỗi

Cho chuỗi S. Hãy đếm số lượng từ trong chuỗi.

Ví dụ: S = “ AB CC DD AA ABC ” → Kết quả: 5

❖ Thuật toán:

- Chuẩn hoá chuỗi S để xoá đi các dấu cách thừa.
- Số lượng từ trong chuỗi = số lượng dấu cách + 1
- Bài toán quy về đếm số lượng dấu cách trong chuỗi.
- Khởi tạo Res = 0;
- Duyệt từ đầu đến cuối chuỗi,

If (s[i] == “ ”)

Res++;

- Đưa Res ra.

❖ **Chương trình tham khảo:**

<pre>#include <iostream> #include <string> using namespace std; string chuanhoa(string s) { string s2 = s; string s1; s1 = " "; size_t pos = s2.find(s1); while (pos != string::npos) { s2.erase(pos, 1); pos = s2.find(s1, pos); } if (s2.size() != 0 && s2[0] == ' ') s2.erase(0, 1); if (s2.size() != 0 && s2.back() == ' ') s2.pop_back(); return s2; } int strcnt(string s, string s1) { int counter = 0; size_t idx = s.find(s1);</pre>	<pre>const fi = 'xau11c2.inp'; fo = 'xau11c2.out'; var s,s1 : string; p,d,i : longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); readln(s); s:=' ' + s; s1:= ' '; while pos(s1,s) > 0 do begin p:= pos(s1,s); delete(s,p,1); end; d:=0; if s[1] = #32 then delete(s,1,1); if s[length(s)] = #32 then delete(s,length(s),10); for i:=1 to length(s) do if s[i] = #32 then inc(d); write(d+1); close(input);close(output); end.</pre>
---	--

<pre> while (idx != string::npos) { counter++; idx = s.find(s1, idx + s1.size()); } return counter; } int main() { string s; cout << "Enter a string: "; getline(cin, s); string s2 = chuanhoa(s); int rent = strcnt(s2, " "); cout << "Result: " << rent + 1; return 0; } </pre>	
--	--

Bài 93. Kí tự đầu tiên các từ in hoa

Chuyển các kí tự đầu tiên của mỗi từ trong xâu thành chữ in hoa.

Ví dụ: S = “ab cc dd aa abc” → Kết quả: S = “Ab Cc Dd Aa Abc”

❖ Thuật toán:

- Chuẩn hoá xâu S để xoá đi các dấu cách thừa trong xâu.
- Đổi S[1] thành kí tự hoa.
- Duyệt từ 2 đến cuối xâu,

Nếu s[i] là dấu cách thì

Đổi s[i+1] thành kí tự hoa;

- Đưa xâu S ra.

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> </pre>	<pre> const fi = 'xau12.inp'; </pre>
--	--

<pre> using namespace std; string chuanhoa(string s) { string s2 = s; string s1; s1 = " "; size_t pos = s2.find(s1); while (pos != string::npos) { s2.erase(pos, 1); pos = s2.find(s1, pos); } if (s2.size() != 0 && s2[0] == ' ') s2.erase(0, 1); if (s2.size() != 0 && s2.back() == ' ') s2.pop_back(); return s2; } int main() { string s; cout << "Enter a string: "; getline(cin, s); s = chuanhoa(s); s = " " + s; for (size_t i = 0; i < s.size() - 1; i++) if (s[i] == ' ') s[i + 1] = toupper(s[i + 1]); s.erase(0, 1); cout << "Result: " << s; return 0; } </pre>	<pre> fo = 'xau12.out'; var s : string; i : longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); read(s); while s[1] = '#32' do delete(s, 1, 1); while s[length(s)] = ' ' do delete(s, length(s), 1); while pos(#32#32, s) > 0 do delete(s, pos(#32#32, s), 1); s[1] := upcase(s[1]); for i:=1 to length(s) do if (s[i] = ' ') then s[i+1]:=upcase(s[i+1]); write(s); close(input);close(output); end. </pre>
--	---

Bài 94. Đếm phân phối

Cho xâu S chỉ gồm các kí tự từ ‘a’ đến ‘z’. Đếm số lần xuất hiện của mỗi kí tự.

Ví dụ: S = “ab cc dd aa abc”

→ Kết quả:

abccddaaabc	a 4
	b 2
	c 3
	d 2

❖ Thuật toán:

Ta sắp xếp lại xâu thành dãy không giảm.

Khởi tạo dem = 1;

Duyệt từ đầu đến cuối xâu,

Nếu $s[i] = s[i+1]$ thì $dem++$;

Ngược lại,

{

cout << s[i] << “ ” << dem;

dem = 1;

}

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <cstring> // using namespace std; std::string ltrssort(std::string s) { size_t n = s.size(); char * cs = new char[n + 1]; strcpy(cs, s.c_str()); bool sorted = false; while (!sorted) { sorted = true; for (size_t i = 0; i < n - 1; i++) {</pre>	<pre>var s,s1:string; i,n,j,d:longint; tam:char; begin assign(input,'x13.inp'); assign(output,'x13.out'); reset(input); rewrite(output); readln(s); n:=length(s); for i:=1 to n-1 do for j:=i+1 to n do begin tam:=s[i]; s[i]:=s[j]; s[j]:=tam;</pre>

```

        if (cs[i] > cs[i + 1])
        {
            char t = cs[i];
            cs[i] = cs[i + 1];
            cs[i + 1] = t;
            sorted = false;
        }
    }
}

s = cs;
delete[] cs;
return s;
}

void distcount(std::string s)
{
    s = ltrssort(s);
    int counter = 1;
    size_t n = s.size();
    for (size_t i = 0; i < n - 1; i++)
    {
        if (s[i] == s[i + 1])
            counter++;
        else
        {
            std::cout << s[i] << " " << counter <<
std::endl;
            counter = 1;
        }
    }
    std::cout << s[n - 1] << " " << counter <<
std::endl;
}

int main()
{
    std::string s;
    std::cout << "Enter a string: ";
        end;
        d:=1;
        for i:=1 to n do
            if s[i]=s[i+1] then inc(d)
            else
                begin
                    writeln(s[i],#32,d);
                    d:=1;
                end;
            close(input);
            close(output);
        end.

```



```
std::cin >> s;
std::cout << "Result:" << std::endl;
distcount(s);
return 0;
}
```

Bài 95. Tách số trong chuỗi

Nhập chuỗi S. Tách các số trong chuỗi ra.

Ví dụ: S = “ab44cb3a2bc”

ab44cb3a2bc	44
	3
	2

❖ Thuật toán:

- Ta tách các số trong chuỗi lưu vào mảng C[k].

Stam = “;” // Khởi tạo chuỗi rỗng

- Duyệt từ đầu đến cuối chuỗi,

Nếu s[i] >= ‘0’ và s[i] <= ‘9’ thì

Nối s[i] vào cuối chuỗi Stam;

Ngược lại,

Đổi chuỗi Stam thành SoStam;

K++;

C[k] = SoStam;

- Duyệt từ 1 tới K,

Đưa c[i] ra.

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <string> using namespace std; int rtmp[100000]; void numseprstr(string s) {</pre>	<pre>const fi = 'xau14c2.inp'; fo = 'xau14c2.out'; var s,t : string; k,i,x : longint; c : array[1..100] of longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output);</pre>

```

string t = "";
int k = 0;
for (size_t i = 0; i < s.size(); i++)
{
    if (s[i] >= '0' && s[i] <= '9')
        t = t + s[i];
    else if (t != "")
    {
        //cout << t << endl;
        rtmp[k] = stoi(t, nullptr, 10);
        k++;
        t = "";
    }
}
if (t.size() > 0)
{
    //cout << t << endl;
    rtmp[k] = stoi(t, nullptr, 10);
    k++;
}
for (int i = 0; i < k; i++)
    cout << rtmp[i] << endl;
}

int main()
{
    string s;
    cout << "Enter string: ";
    getline(cin, s);
    cout << "Result:" << endl;
    numseprstr(s);
    return 0;
}

```

```

readln(s);
k:=0 ;
s:= s + ' ';
s:= ' ' + s;
for i:=1 to length(s) do
    if s[i] in ['0'..'9'] then
        t := t + s[i]
    else
        if t <> " then
            begin
                val(t,x);
                k := k+1;
                c[k] := x;
                t := "";
            end;
        for i:=1 to k do
            write(c[i],#32);
close(input);close(output);
end.

```

Bài 96. Số lớn nhất trong xâu

Nhập xâu S. Tìm số lớn nhất trong xâu.

Ví dụ: S = “ab44cb3a2bc”

--	--

ab44cb3a2bc

44

❖ Thuật toán:

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <string> using namespace std; int rtmp[100000]; void numfmaxstr(string s) { string t = ""; int k = 0; for (size_t i = 0; i < s.size(); i++) { if (s[i] >= '0' && s[i] <= '9') t = t + s[i]; else if (t != "") { //cout << t << endl; rtmp[k] = stoi(t, nullptr, 10); k++; t = ""; } } if (t.size() > 0) { //cout << t << endl; rtmp[k] = stoi(t, nullptr, 10); k++; } if (k > 0) { int rmax = rtmp[0]; for (int i = 1; i < k; i++) </pre>	<pre> uses math; const fi = 'xau15c2.inp'; fo = 'xau15c2.out'; var s,t : string; k,i,x,rmax : longint; c : array[1..100] of longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); readln(s); k:=0 ; s:= ' ' + s + ' '; for i:=1 to length(s) do if s[i] in ['0'..'9'] then t := t + s[i] else if t <> " then begin val(t,x); k := k+1; c[k] := x; t := ""; end; for i:=1 to k do rmax:=max(rmax,c[i]); write(rmax); close(input);close(output); end. </pre>
---	--

<pre> if (rtmp[i] > rmax) rmax = rtmp[i]; cout << rmax; } } int main() { string s; cout << "Enter string: "; getline(cin, s); cout << "Result:" << endl; numfmaxstr(s); return 0; } </pre>	
--	--

Bài 97. Sắp xếp kí tự

Sắp xếp các kí tự trong xâu thành xâu không giảm.

Ví dụ:

abccddaaabc	aaaabbccdd

❖ Thuật toán:

- Áp dụng giải thuật sắp xếp nổi bọt để sắp xếp xâu thành dãy không giảm.

- For(i, 0, s.size()-2)

For(j, i+1, s.size()-1)

If (s[i] > s[j])

Swap(s[i], s[j]); // đổi giá trị s[i] với s[j]

❖ Chương trình tham khảo:

<pre> #include <iostream> #include <cstring> // using namespace std; std::string ltrssort(std::string s) { </pre>	<pre> const fi = 'xau17.inp'; fo = 'xau17.out'; var s,k : string; t : char; i,l,j: longint; </pre>
---	--

<pre> std::size_t n = s.size(); char * cs = new char[n + 1]; strcpy(cs, s.c_str()); for (size_t i = 0; i < n - 1; i++) for (size_t j = i + 1; j < n; j++) if (cs[i] > cs[j]) std::swap(cs[i], cs[j]); s = cs; delete[] cs; return s; } int main() { std::string s; std::cout << "Enter string: "; std::getline(std::cin, s); std::cout << "Result: " << ltrssort(s); return 0; } </pre>	<pre> //c : array[1..100] of char; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); read(s); l := length(s); for i:=1 to l-1 do for j:=i+1 to l do if s[i] > s[j] then begin t:=s[i]; s[i]:=s[j]; s[j]:=t; end; write(s); end; close(input);close(output); end. </pre>
---	--

Bài 98. Đếm kí tự 1

Nhập vào S. Đếm số lượng kí tự ‘a’ trong S.

Ví dụ:

abccddaaabc	4

❖ Thuật toán:

- Khởi tạo res = 0.
- Duyệt từ đầu đến cuối S,
 - Nếu S[i] = ‘a’ thì
 - Tăng res lên 1 đơn vị.
- Trả về res.

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; void submain(string s) { int counter = 0; for (size_t i = 0; i < s.size(); i++) if (s[i] == 'a') counter++; cout << "Result: " << counter; } int main() { string s; cin >> s; submain(s); return 0; } </pre>	<pre> const fi = 'xau18.inp'; fo = 'xau18.out'; var s : string; i,d,l: longint; //c : array[1..100] of char; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); read(s); l := length(s); d:=0; for i:=1 to l do if s[i] = 'a' then inc(d); write(d); close(input);close(output); end. </pre>
---	---

Bài 99. Nén xâu

Nhập xâu S. Hãy nén xâu S. Đưa xâu nén ra.

Ví dụ: S= 'aaabbcccd' → S_{nén} = 'a3b3c4d1'

❖ Thuật toán:

Duyệt từ đầu đến cuối xâu,

Nếu s[i] = s[i+1] thì tăng đếm lên

Ngược lại,

```

{
    Lưu đếm vào mảng c[k];
    Lưu s[i] vào xâu kết quả Sa;
    Khởi tạo lại đếm = 1;
}

```

Duyệt từ 1 tới k,

Đưa (sa[i],c[i]) ra.

❖ **Chương trình tham khảo:**

<pre> #include <iostream> using namespace std; char b[100000]; int c[100000]; void submain(string s) { int counter = 1; int k = 0; size_t n = s.size(); for (size_t i = 0; i < n - 1; i++) { if (s[i] == s[i + 1]) counter++; else { b[k] = s[i]; c[k] = counter; k++; counter = 1; } } b[k] = s[n - 1]; c[k] = counter; k++; for (int i = 0; i < k; i++) cout << b[i] << c[i]; } int main() { string s; </pre>	<pre> const fi = 'xau19.inp'; fo = 'xau19.out'; var s,a : string; i,d,l,k: longint; c : array[1..100] of longint; begin assign(input,fi);reset(input); assign(output,fo);rewrite(output); read(s); a:=' '; s:=s+' '; l := length(s); d:=1; for i:=1 to l do if s[i] = s[i+1] then inc(d) else if s[i] <> s[i+1] then begin a:=a+s[i]; inc(k); c[k]:=d; d:=1; end; for i:=1 to k do write(a[i],c[i]); close(input);close(output); end. </pre>
---	--

<pre>cout << "Enter string: "; getline(cin, s); submain(s); return 0; }</pre>	
---	--

Bài 100. Giải nén xâu.

Nhập xâu Snen. Hãy giải nén xâu.

Ví dụ: Snen = 'a3b3c4d1' → S = 'aaabbcccd'

❖ Thuật toán:

- Thực hiện tách số ra khỏi xâu lưu vào mảng C[k], kí tự lưu vào mảng D[k].

+ Stam = ''; //Khởi tạo xâu rỗng

+ Duyệt từ đầu đến cuối xâu,

Nếu s[i] >= '0' và s[i] <= '9' thì

Nối s[i] vào cuối xâu Stam;

Ngược lại,

Nếu Stam khác rỗng,

Đổi xâu Stam thành SoStam;

K++;

C[k] = SoStam;

D[k] = s[i];

Ngược lại,

i++;

- Duyệt từ 1 tới K,

Với mỗi kí tự d[i] ta đưa ra C[i] lần.

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <string> using namespace std; char c[100000]; int rtmp[100000];</pre>	<pre>const fi = 'xau20.inp'; fo = 'xau20.out'; var s,a,res,t : string; i,j,l,x,k : longint; c : array[1..100] of longint;</pre>
---	--


```

string res;

void submain(string s)
{
    string t = "";
    size_t i = 0;
    int k = 0;
    while(i < s.size())
    {
        if (s[i] >= 'a' && s[i] <= 'z') c[k] = s[i];
        while (s[i] >= '0' && s[i] <= '9')
        {
            t = t + s[i];
            i++;
        }
        if (t != "")
        {
            //cout << t << endl;
            rtmp[k] = stoi(t, nullptr, 10);
            k++;
            t = "";
        }
        else
            i++;
    }
    if (t.size() > 0)
    {
        //cout << t << endl;
        rtmp[k] = stoi(t, nullptr, 10);
        k++;
    }
    for (int i = 0; i < k; i++)

```

```

begin
    assign(input,fi);reset(input);
    assign(output,fo);rewrite(output);
    readln(s);
    l := length(s);
    t:="";
    i:=1;
    k:=0;
    while i <= l+1 do
        begin
            if s[i] in ['0'..'9'] then
                begin
                    t:=t+s[i];
                    inc(i);
                end
            else
                if t <> " then
                    begin
                        val(t,x);
                        inc(k);
                        c[k]:= x;
                        t:= "";
                    end
                else
                    begin
                        a:=a + s[i];
                        inc(i);
                    end;
                end;
            res:="";
            for i:=1 to k do
                begin
                    for j:=1 to c[i] do

```

<pre> for (int j = 0; j < rtmp[i]; j++) cout << c[i]; } int main() { res = ""; string s; cout << "Enter string: "; cin >> s; submain(s); return 0; } </pre>	<pre> res:=res+a[i]; end; write(res); close(input);close(output); end. </pre>
---	--

Bài 101. Xóa S1 trong S

Nhập vào S và S1. Hãy xóa tất cả các S1 trong S.

Đưa vào S sau khi xóa ra.

Ví dụ: S = "abcbabc"; S1 = "bc" → Kết quả: S = "aba";

❖ Thuật toán:

Sử dụng hàm Pos(s1, s) để tìm vị trí đầu tiên của S1 trong S, xóa đi một ký tự trong S từ vị trí tìm thấy và lặp lại đến khi không còn S1 trong S.

❖ Chương trình tham khảo:

<pre> #include <iostream> using namespace std; string submain(string s, string s1) { string s2 = s; size_t pos = s2.find(s1); while (pos != string::npos) { s2.erase(pos, s1.size()); pos = s2.find(s1, pos); } } </pre>	<pre> var s ,s1 : String; p : Longint; begin readln(s1); readln(s); while (pos(s1, s)>0) and (s <> "") do begin p := pos(s1, s); delete(s,1,p); end; write(s); end. </pre>
--	---

<pre> return s2; } int main() { string s, s1; cout << "Enter first string: "; cin >> s; cout << "Enter second string: "; cin >> s1; cout << "Result: " << submain(s, s1); return 0; } </pre>	
---	--

Bài 102. Xâu Fibonacci 1

$F[1]='a'$; $F[2]='b'$; $F[n] = F[n-1] + F[n-2]$.

Nhập N. Hãy tìm xâu Fibonacci thứ N.

❖ **Thuật toán:**

❖ **Chương trình tham khảo:**

<pre> #include <iostream> #include <string> using namespace std; int main() { string fm = "b", fm1 = "a", fm2 = ""; int n; cin >> n; for (int i = 3; i <= n; i++) { fm2 = fm1; fm1 = fm; fm = fm1 + fm2; } cout << fm; } </pre>	
--	--

<pre>return 0; }</pre>	
------------------------	--

Bài 103. Xâu Fibonacci 2

$F[1]='a'$; $F[2]='b'$; $F[n] = F[n-1] + F[n-2]$.

Nhập N. Hãy đếm số kí tự 'a' và kí tự 'b' trong xâu Fibonacci thứ N.

❖ Thuật toán:

- Ta $F[1] = 'a'$; $F[2] = 'b'$.

- Duyệt từ 3 tới N,

$F[i] = F[i-1] + F[i-2]$; //phép toán nối xâu

- Đếm số lượng kí tự 'a' và kí tự 'b' trong $F[n]$:

Duyệt đầu đến cuối xâu $F[n]$,

Nếu $s[i] = 'a'$ thì $resa++$

Ngược $resb++$;

- Đưaa $resa$, $resb$ ra.

* Tiếp cận theo hướng qui hoạch động:

Gọi $G[i]$ là số lượng kí tự 'a' trong xâu $F[i]$.

Ta có, $G[1] = 1$; $G[2] = 0$; $G[i] = G[i-1] + G[i-2]$;

Tương tự, ta gọi $H[i]$ là số lượng kí tự 'b' trong xâu $F[i]$.

Ta có, $H[1] = 0$; $H[2] = 1$; $H[i] = H[i-1] + H[i-2]$;

❖ Video bài giảng: <https://youtu.be/4kSfLQGWfQ4>

❖ Chương trình tham khảo:

<pre>#include <iostream> #define NMAX 100001 using namespace std; long long ca[NMAX], cb[NMAX]; int main() { int n; cin >> n; ca[1] = 1; ca[2] = 0;</pre>	
--	--

<pre> cb[1] = 0; cb[2] = 1; for (int i = 3; i <= n; i++) { ca[i] = ca[i - 1] + ca[i - 2]; cb[i] = cb[i - 1] + cb[i - 2]; } cout << "Result: " << ca[n] << " " << cb[n]; return 0; } </pre>	
---	--

Bài 104. Xâu con palidrome

Xâu đối xứng (Palindrome) là xâu mà đọc từ trái sang phải và từ phải sang trái ta được cùng một xâu. Ví dụ các xâu Palindrome: ‘a’, ‘aba’, ‘abcba’, ‘1’, ‘121’, ...

Cho xâu S. Tìm độ dài của xâu con liên tiếp dài nhất là xâu palidrome.

Ví dụ: S = “abcbabc” → Kết quả: Res = 5; và S = “abcba”;

❖ **Thuật toán:**

❖ **Chương trình tham khảo:**

<pre> #include <iostream> using namespace std; string str_reverse(string s) { int n = s.size(); string s1; s1 = ""; for (int i = 0; i < n; i++) { s1 = s[i] + s1; } return s1; } bool palin(string s) { </pre>	
--	--

```

    return s == str_reverse(s);
}

string xuli(string s)
{
    string res = "";
    string t = "";
    size_t sz = s.size();
    for (size_t i = 0; i < sz; i++)
    {
        bool found = false;
        for (size_t j = sz - 1; j >= i; j--)
        {
            t = s.substr(i, j - i + 1);
            if (palin(t))
            {
                res = t;
                found = true;
                break;
            }
        }
        if (found)
            break;
    }
    return res;
}

int main()
{
    string s;
    cout << "Enter a string: ";
    getline(cin, s);
    cout << "Result: " << xuli(s);
    return 0;
}

```

Bài 105. Sắp xếp số trong chuỗi

Cho chuỗi S. Hãy sắp xếp các số trong chuỗi thành dãy không giảm.

Ví dụ: S='ab12c3d4' → Smoi ='ab3c4d12'.

❖ **Thuật toán:**

- Thực hiện tách phần số lưu vào mảng C[k] và phần chữ lưu vào mảng D[k].
- Sắp xếp mảng C[k] thành dãy không giảm.
- Duyệt từ 1 tới K,

Đưa (D[i] và C[i]) ra.

- Lưu ý, cần xử lý trường hợp tương tự như S = '12abc3de', S = '12abd43d'

❖ **Chương trình tham khảo:**

<pre> #include <iostream> #include <string> using namespace std; string c[10000]; int d[10000]; void nsort(int n) { for (int i = 0; i < n - 1; i++) for (int j = i + 1; j < n; j++) if (d[i] > d[j]) std::swap(d[i], d[j]); } void numseprstr(string s) { int nc = 0, nd = 0; string t = ""; bool p = s[0] >= '0' && s[0] <= '9'; for (size_t i = 0; i < s.size(); i++) { if (s[i] >= '0' && s[i] <= '9') t = t + s[i]; else if (t != "") { //cout << t << endl; d[nd] = stoi(t, nullptr, 10); } } } </pre>	
---	--

```
        nd++;
        t = "";
    }
}
if (t.size() > 0)
{
    //cout << t << endl;
    d[nd] = stoi(t, nullptr, 10);
    nd++;
}
t = "";
for (size_t i = 0; i < s.size(); i++)
{
    if (s[i] < '0' || s[i] > '9')
        t = t + s[i];
    else if (t != "")
    {
        //cout << t << endl;
        c[nc] = t;
        nc++;
        t = "";
    }
}
if (t.size() > 0)
{
    //cout << t << endl;
    c[nc] = t;
    nc++;
}
nsort(nd);
int i1 = 0, i2 = 0;
while (i1 < nc && i2 < nd)
{
    if (p)
    {
        cout << d[i2];
        i2++;
    }
}
```



```

    else
    {
        cout << c[i1];
        i1++;
    }
    p = !p;
}
while (i2 < nd)
{
    cout << d[i2];
    i2++;
}
while (i1 < nc)
{
    cout << c[i1];
    i1++;
}

}

int main()
{
    string s;
    cout << "Enter string: ";
    getline(cin, s);
    cout << "Result:" << endl;
    numseprstr(s);
    return 0;
}

```

Bài 106. Mã hóa chuỗi

Mã hóa chuỗi S chỉ gồm các ký tự từ a tới z, bằng cách dịch phải các ký tự theo vòng tròn đi K ký tự trên bảng chữ cái.

Cho chuỗi S và K. Tìm chuỗi mã hóa theo cách dịch phải như trên.

Ví dụ: S = "abcbabcz"; K = 1 → Kết quả: S = "bcdcbceda";

❖ Thuật toán:

- Thực hiện mã hoá bằng cách lấy số thứ tự của s[i] trong bảng mã ASCII rồi cộng thêm K.

- Vì là mã hoá vòng, nên khi kết quả của phép toán trên lớn hơn 122 (Stt của kí tự 'z') thì ta phải chia dư cho 26 (số lượng kí tự từ 'a' tới 'z').

$$s[i] = (\text{char})((s[i] - 0 + k) \% 26 + 78);$$

❖ **Chương trình tham khảo:**

<pre> #include <iostream> #include <string> using namespace std; void xuli(string s, int k) { size_t n = s.size(); for (size_t i = 0; i < n; i++) { if (s[i] >= 'a' && s[i] <= 'z') { s[i] = (char)((s[i] - 0 + k) % 26 + 78); } } cout << "Result: " << s; } int main() { string s; int k; cout << "Enter string: "; getline(cin, s); cout << "Enter a number: "; cin >> k; xuli(s, k); return 0; } </pre>	
---	--

Bài 107. Giải mã xâu

Cho xâu mã hóa bằng cách dịch phải K kí tự. Nhập S và K. Hãy giải mã tìm xâu S ban đầu.

Ví dụ: S = “bcdebcda”; K = 1 → Kết quả: S = “abcbabcz”;

❖ **Thuật toán:**

Gọi p số thứ tự của s[i] trong bảng mã ASCII, nếu $p - k \geq 97$ thì s[i] là kí tự thứ $p - k$.

Ngược lại, do mã hoá vòng nên s[i] là kí tự thứ $p - k + 26$.

❖ **Chương trình tham khảo:**

<pre> #include <iostream> #include <string> using namespace std; void xuli(string s, int k) { size_t n = s.size(); for (size_t i = 0; i < n; i++) { if (s[i] >= 'a' && s[i] <= 'z') { int tmp = s[i] - 0 - k; if (tmp >= 97) s[i] = (char)tmp; else s[i] = (char)(tmp + 26); } } cout << "Result: " << s; } int main() { string s; int k; cout << "Enter string: "; getline(cin, s); cout << "Enter a number: "; cin >> k; xuli(s, k); </pre>	
--	--

```
return 0;
}
```

Bài 108. Ghép xâu số

Cho N xâu số s_1, s_2, \dots, s_N . Hãy ghép các xâu này để được số lớn nhất.

Ví dụ:

3	931228
9 28 312	

❖ Thuật toán:

Thực hiện sắp xếp dãy số thành dãy giảm dần theo tiêu chí $\text{sosanh}(a[i], a[j]) = \text{true}$ nếu ghép $a[i]$ trước $a[j]$ cho kết quả lớn hơn ghép $a[j]$ trước $a[i]$.

❖ Chương trình tham khảo:

<pre>#include <iostream> #include <string> #define NMAX 100000 using namespace std; int a[NMAX]; size_t n; bool compar(int u, int v) { string su = to_string(u); string sv = to_string(v); return su < sv; } void xuli() { for (size_t i = 0; i < n - 1; i++) for (size_t j = i + 1; j < n; j++) if (compar(a[i], a[j])) swap(a[i], a[j]); }</pre>	

<pre> for (size_t i = 0; i < n; i++) cout << a[i]; } int main() { cout << "Enter number of elements: "; cin >> n; cout << "Enter " << n << " elements: " << endl; for (size_t i = 0; i < n; i++) cin >> a[i]; xuli(); return 0; } </pre>	
---	--

Bài 109. Xâu con chung dài nhất

Cho xâu S1 và S2. Tìm độ dài xâu con chung liên tiếp dài nhất của hai xâu.

abccba	2
baac	//”ba”

<pre> #include <iostream> #include <string> #define NMAX 3001 using namespace std; size_t f[NMAX][NMAX]; void xuli(string s1, string s2) { f[0][0] = 0; f[0][1] = 0; f[1][0] = 0; s1 = " " + s1; s2 = " " + s2; </pre>	
---	--

```

size_t n1 = s1.size();
size_t n2 = s2.size();
size_t res = 0;
for (size_t i = 1; i <= n1; i++)
    for (size_t j = 1; j <= n2; j++)
    {
        if (s1[i] == s2[j])
            f[i][j] = f[i - 1][j - 1] + 1;
        else
            f[i][j] = 0;
        res = max(res, f[i][j]);
    }
cout << "Result: " << res;
}

int main()
{
    string s1, s2;
    cout << "Enter first string: ";
    getline(cin, s1);
    cout << "Enter second string: ";
    getline(cin, s2);
    xuli(s1, s2);
    return 0;
}

```

Bài 110. Nhầm lẫn

MrT làm các phép toán cộng, phép toán trừ rất tốt, chỉ có một tội: rất hay chép nhầm đầu bài và cũng chỉ nhầm ở hai số 5, 6: đôi khi anh ấy chép nhầm 5 thành 6 hoặc ngược lại. Ví dụ số 25 có thể ghi thành số 26.

Như vậy, nếu đầu bài cho là $11 + 25$ thì MrT có thể có kết quả là 36 hoặc 37.

Yêu cầu: Cho 2 số nguyên dương **a** và **b** ($1 \leq a, b \leq 10^{12}$). Hãy xác định kết quả nhỏ nhất và lớn nhất mà MrT có được.

Dữ liệu: Vào từ tệp MISTAKE.INP gồm một dòng chứa 2 số nguyên dương a và b.

Kết quả: Đưa ra file MISTAKE.OUT trên một dòng 2 số nhỏ nhất và lớn nhất có thể nhận được.

Ví dụ:

MISTAKE.INP	MISTAKE.OUT
11 25	36 37

---Hết---