



KỸ THUẬT LẬP TRÌNH

Bài tập 3 Các cấu trúc lặp

Bài tập bắt buộc

Bài 1. Viết chương trình cho phép người dùng nhập vào hai số nguyên dương a và n . Sử dụng cấu trúc lặp đã học để tính a lũy thừa n (a^n).

Bài 2. Viết chương trình cho phép người dùng nhập vào số nguyên. Kiểm tra n có phải là số nguyên tố hay không ?

Bài 3. Viết chương trình tính tổng S như sau:

Với N là số nguyên dương.
$$S = 1 - \frac{1}{1 \times 2} + \frac{1}{2 \times 3} - \frac{1}{3 \times 4} + \dots + \frac{(-1)^N}{N \times (N+1)}$$

Bài 4. Sử dụng cấu trúc lựa chọn và cấu trúc lặp đã học thực hiện các yêu cầu sau:

- Cho phép người dùng nhập vào lần lượt từng số thực. Việc nhập dữ liệu của người dùng sẽ kết thúc khi người dùng nhập vào số 0.
- Hãy cho biết người dùng đã nhập vào bao nhiêu số thực (kể cả số 0) và số thực nhỏ nhất trong những số thực do người dùng nhập vào (kể cả số 0).

Bài 5.

Viết chương trình nhập vào hai số nguyên dương m và n . Kiểm tra xem m và n có phải là hai số nguyên tố cùng nhau hay không?

Gợi ý: m và n là nguyên tố cùng nhau khi ước số chung lớn nhất của chúng là 1.

Bài 6. Viết chương trình nhập vào số nguyên dương N . Tính tổng các ước số của N (không kể 1 và N).

Bài 7. Viết chương trình xuất ra ma trận bên dưới. Chỉ cần xuất ra số và khoảng cách giữa các số, không cần xuất ra đường kẻ ma trận.



1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1

Bài tập nâng cao

Bài 8.

Viết chương trình nhập vào số nguyên dương N, sau đó tính tổng các nghịch đảo của bình phương tất cả các số tự nhiên từ 1 đến N.

$$S = \sum_{i=1}^N \frac{1}{i^2}$$

Bài 9. Viết chương trình cho phép người dùng nhập vào số nguyên dương, xuất ra số nguyên tố thứ n.

Bài 10. Phân tích một số nguyên dương thành tích các thừa số nguyên tố.

Bài 11. Nhập một số nguyên dương n. Tìm số hàng chục và hàng trăm của n!.

Bài 12 .

Giới thiệu

Giả sử chúng ta đã có một số hàm thư viện xây dựng sẵn với prototype như sau:

```
int read (int x, int y)
int replace (int x, int y, int old_color, int new_color)
```



Khoa Khoa học và Kỹ thuật Máy tính
Bộ môn Khoa học Máy tính

Các hàm này dùng để vẽ và xử lý một file hình ảnh. Một file hình ảnh sẽ được biểu diễn thành một ma trận hai chiều như sau:

`int pic[n][m]`

trong đó mỗi phần tử $pic[x,y]$ của ma trận sẽ là giá trị màu của điểm ảnh trên hình ở tọa độ x, y . Mỗi màu sẽ được biểu diễn bằng một giá trị số được định nghĩa như sau:

WHITE 1

BLACK 2

RED 3

GREEN 4

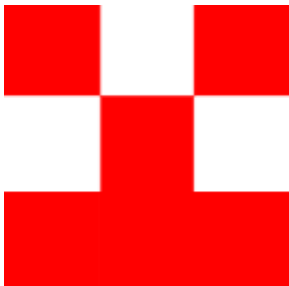
BLUE 5

YELLOW 6

Ví dụ bên dưới cho thấy một hình ảnh thật như sau:



Hình ảnh này phóng lớn bao gồm 9 điểm ảnh, tạo thành ma trận 3 x 3 như sau



Như vậy ma trận pic trong trường hợp này sẽ là một ma trận 3×3 như sau

$$\begin{bmatrix} 3 & 1 & 3 \\ 1 & 3 & 1 \\ 3 & 3 & 3 \end{bmatrix}$$

Yêu cầu

Với đoạn mã như sau:

`int pic[n][m];`



Khoa Khoa học và Kỹ thuật Máy tính
Bộ môn Khoa học Máy tính

//INIT...

//TODO: write your code here

//RE-WRITE:

Đoạn mã chú thích INIT đã được viết sẵn để đọc dữ liệu từ một file hình ảnh thật và chép dữ liệu và ma trận *pic*. Đoạn mã chú thích RE-WRITE sẽ chép dữ liệu từ ma trận *pic* vào hình ảnh thật.

Hãy viết đoạn mã cho phần chú thích TODO để làm các công việc sau:

- a. Kiểm tra xem trong hình gốc có điểm ảnh nào màu đỏ hay không
- b. Thay tất cả điểm ảnh màu đỏ và màu xanh trong hình gốc thành màu trắng

HẾT