

Bài 8: Test Program: Lớn hơn hay nhỏ hơn

Test Program: Lớn hơn hay nhỏ hơn, chương trình đầu tiên của bạn

Bây giờ chúng ta đến phần Test Program đầu tiên. Phần này sẽ kiểm chứng xem bạn có thể thực hiện lại những gì bạn đã học được không. Bởi vì, lý thuyết thì chỉ có vậy nhưng nếu chúng ta không biết cách thực hành như thế nào thì thật uổng phí thời gian bạn bỏ ra để học chúng. 😊

Bạn có tin không? Nếu bạn đã đến được đây có nghĩa là bạn đã có đủ khả năng tạo ra một chương trình khá thú vị rồi đây.

Một trò chơi trên console. Khá đơn giản, vì thế tôi chọn nó làm đề bài Test Program đầu tiên.

Chuẩn bị và một vài gợi ý

Nguyên tắc của chương trình

Trước tiên, tôi phải giải thích các bạn chương trình của chúng ta hoạt động dựa vào đâu.

Đây là một trò chơi gọi là «Lớn hơn hay nhỏ hơn»

Nguyên tắc là như sau:

1. Máy tính sẽ chọn ngẫu nhiên một số từ 1 đến 100.
2. Máy tính sẽ yêu cầu bạn đoán số đó là bao nhiêu.
3. Máy tính sẽ so sánh số bạn chọn và số « bí mật » đó. Máy tính sẽ gợi ý cho bạn biết số bí mật này lớn hay nhỏ hơn số mà bạn đã chọn. Nếu đoán đúng thì chương trình dừng lại.
4. Nếu bạn đoán sai ở bước 3, thì máy tính lại kêu bạn đoán tiếp.
5. ... Và máy tính lại gợi ý số bí mật lớn hơn hay nhỏ hơn.
6. Và nó vẫn tiếp tục cho đến khi bạn tìm thấy số bí mật đó.

Mục đích của trò chơi đương nhiên là tìm số bí ẩn với số lần đoán nhỏ nhất 😊

Và đây là màn hình của một phần chơi, và đây là những gì bạn phải làm :

Console:

```
So can tim la bao nhieu ? 50
Lon hon !
So can tim la bao nhieu ? 75
Lon hon !
So can tim la bao nhieu ? 85
Nho hon !
So can tim la bao nhieu ? 80
Nho hon !
So can tim la bao nhieu ? 78
Lon hon !
So can tim la bao nhieu ? 79
Chuc mung ! Ban da tim duoc so bi mat !!!
```

Cách chọn ngẫu nhiên một số



Vậy làm sao để máy tính chọn ngẫu nhiên một số? Tôi vẫn chưa biết làm như thế nào !

Và. 😞

Đến lúc này, bạn vẫn chưa biết cách yêu cầu máy tính chọn ra một số ngẫu nhiên. Tuy máy tính tính rất giỏi việc tính toán nhưng nó không biết cách nào để thực hiện công việc trên nên điều này không hề đơn giản. 😞

Trên thực tế, để « thử » có được một số ngẫu nhiên, người ta phải yêu cầu máy tính thực hiện những phép toán khá phức tạp...

Ta có 2 cách sau:

- Ta yêu cầu người sử dụng nhập vào một số ngẫu nhiên với hàm scanf trước. Điều này dẫn đến phải có hai người cùng chơi trò chơi này: một người nhập vào số bí mật ngẫu nhiên và người còn lại đoán số bí mật đó.
- Hoặc là ta cố gắng khiến máy tính tự tạo ra một số ngẫu nhiên. Lợi thế là chúng ta có thể chơi ngay trò này một mình. Bất lợi... là tôi phải hướng dẫn các bạn làm cách nào để thực hiện 😞

Chúng ta sẽ chọn cách thứ hai, và tôi không hề ngăn cản bạn code cách thứ nhất nếu như bạn muốn. 😊

Để tạo ra một số ngẫu nhiên, người ta sử dụng function `rand()`. Function này sẽ tạo ra một số ngẫu nhiên bất kì. Nhưng chúng ta lại cần một số ngẫu nhiên từ 1 đến 100. (Vì nếu chúng ta không biết được giới hạn của số bí mật, thì trò chơi sẽ vô cùng khó). 😊

Để làm được những việc trên, chúng ta sẽ sử dụng công thức sau (tất nhiên tôi sẽ giải thích cho bạn ý nghĩa của nó là gì ! 🤖):

Code C:

```
srand (time(NULL));  
soBiMat = (rand( ) % (MAX - MIN + 1)) + MIN;
```

Dòng thứ nhất (với hàm `srand`) nó có tác dụng hỗ trợ cho các hàm random bắt đầu công việc khởi tạo một số ngẫu nhiên. Vâng, hơi phức tạp một tí, tôi đã báo trước cho bạn rồi, `soBiMat` là một biến số chứa số được chọn ra ngẫu nhiên.



Instruction srand phải được chạy duy nhất một lần (ở đầu chương trình). Bắt buộc chỉ một lần, duy nhất một lần. Tác dụng của dòng `srand (time(NULL))`; là để giúp chương trình tránh gặp phải trường hợp chọn trùng 2 số ngẫu nhiên (trường hợp này rất hi hữu nhưng nó vẫn xảy ra, và ta phải tránh nó)

Sau đó bạn có thể sử dụng `rand()` bao nhiêu lần tùy thích nhưng đừng quên rằng không để máy tính thực hiện `srand` nhiều hơn hai lần trong một chương trình.

MAX và MIN là những biến số constants, số đầu tiên là giới hạn lớn nhất (100) và số thứ hai là giới hạn nhỏ nhất (1). Tôi khuyên bạn nên xác định những constants này ở đầu chương trình. Như sau:

Code C:

```
const int MAX = 100, MIN = 1;
```

Những thư viện kèm theo

Để chương trình bạn không bị lỗi, chúng ta cần phải thêm vào ba thư viện: `stdlib`, `stdio` và `time` (cái cuối cùng dùng cho những số ngẫu nhiên).

Và chương trình phải được bắt đầu bởi:

Code C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

Tôi chỉ nói nhiều đó thôi !

Tôi sẽ không giải thích thêm cho bạn vì nếu tiếp tục tôi sẽ lộ ra hết toàn bộ code của chương trình ! 😊



Để bạn có thể tạo được những số ngẫu nhiên, tôi bắt buộc chỉ cho bạn những dòng codes « ăn liền » ở trên và không giải thích rõ nó hoạt động như thế nào. Tôi thực sự không thích điều này lắm, nhưng vì tôi không còn lựa chọn nào khác trong thời điểm hiện tại. Sau này tôi sẽ chỉ bạn những kiến thức mới giúp bạn hiểu được những vấn đề trên. 😊

Tóm lại, bạn đã có đủ kiến thức, tôi cũng đã giải thích cho bạn nguyên tắc của trò chơi, tôi đã cho bạn thấy một phần màn hình trò chơi của chương trình.

Với nhiều đó thông tin, tôi nghĩ bạn đã có thể tự mình viết chương trình này.

Nào hãy bắt đầu thôi !

Chúc bạn may mắn ! 😊

Đáp án Test Program "Lớn hơn hay Nhỏ hơn"

Đây là đáp án tôi đưa ra cho bạn:

C Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char** argv)
{
    int soBiMat = 0, soHienTai = 0;
    const int MAX = 100, MIN = 1;
    // Tao mot so ngau nhien
    srand(time(NULL));
    soBiMat = (rand() % (MAX - MIN + 1) + MIN);
    /*Vong lap cua chuong trinh. No se tu lap lai cho den khi nguoi
    choi tim duoc so bi mat*/
    do
    {
        //Yeu cau doan so bi mat
        printf("So can tim la bao nhieu ? ");
        scanf("%d", &soHienTai);
        //So sanh so hien tai voi so bi mat
        if (soBiMat > soHienTai)
            printf("Lon hon !\n");
        else if (soBiMat < soHienTai)
            printf("Nho hon !\n");
        else
            printf("Chuc mung ! Ban da tim duoc so bi mat !!!\n");
    } while (soHienTai != soBiMat);

    return 0;
}
```

Giải thích

Bây giờ tôi sẽ giải thích cho bạn đoạn code của tôi, chúng ta bắt đầu từ dòng đầu tiên.

Preprocessor directives

(những chỉ thị tiền xử lý)

Đó là những dòng bắt đầu bằng # ở đầu đoạn code. Nó tích hợp các thư viện ta cần vào chương trình. Tôi vừa đưa luôn 3 dòng này cho bạn ở phần Test Program, nếu bạn có lỗi ở chỗ này, khả năng của bạn thật sự bá đạo. 😊

Những biến số

Chúng ta không cần nhiều lắm. Chỉ một biến số để chứa số mà người dùng đoán (*soHienTai*) và một biến số khác chứa số bí mật được chọn ngẫu nhiên bởi máy tính (*soBiMat*).

Tôi cũng khai báo những *constants* mà tôi giới thiệu với bạn ở đầu chương hướng dẫn này. Lợi ích của việc khai báo *constant* ở đầu chương trình, là bạn có thể thay đổi độ khó của trò chơi dễ dàng (ví dụ *MAX* = 1000). Chúng ta đơn giản chỉ cần thay đổi giá trị của dòng này và dịch lại chương trình.

Vòng lặp

Tôi chọn vòng lặp *do...while*. Về mặt lý thuyết, vòng lặp *while* cũng hoạt động, nhưng tôi thấy rằng ý tưởng *do...while* sẽ có tính logic hơn.

Tại sao? Bởi vì, bạn hãy nhớ lại xem, *do...while* là vòng lặp chạy ít nhất một lần. Và chúng ta cũng yêu cầu người chơi đoán số bí mật ít nhất một lần (người chơi không thể nào có được kết quả nếu chưa đoán lần nào, làm được thì là thánh rồi!). 😊

Mỗi lần vòng lặp hoạt động, ta sẽ yêu cầu người chơi đoán một số. Ta sẽ đặt số đó vào biến số *soHienTai*. Sau đó so sánh với biến số *soBiMat*. Có 3 khả năng xảy ra:

1. *soBiMat* lớn hơn *soHienTai*, sẽ thông báo "Lớn hơn !".
2. *soBiMat* nhỏ hơn *soHienTai*, sẽ thông báo "Nhỏ hơn !"
3. Vậy nếu *soBiMat* không lớn hơn cũng không nhỏ hơn *soHienTai* ? Bằng nhau là điều chắc chắn ! ở vị trí **else**. Tại trường hợp này, thông báo "Chúc mừng ! Bạn đã tìm được số bí mật !!!"

Cần phải có một điều kiện cho vòng lặp. Khá đơn giản để nhận ra: vòng lặp vẫn còn lặp lại hễ *soHienTai* khác với *soBiMat*. Nếu 2 số bằng nhau, đồng nghĩa với người chơi đã tìm ra đáp án, vòng lặp ngừng. Chương trình kết thúc. 🎉

Ý tưởng cải tiến

Bạn tưởng phần Test Program kết thúc ở đây à? Tôi muốn bạn tiếp tục cải tiến chương trình, đồng thời tập luyện. Đừng bao giờ quên rằng chỉ có luyện tập thường xuyên mới có thể khiến bạn ngày càng tiến bộ! Với những ai đọc bài hướng dẫn của tôi mà không test lại một lần nào thì đó là một sai lầm rất lớn, tôi đã từng nói điều này và tôi sẽ tiếp tục nói lại đây!

Báo cho bạn biết đầu tôi lúc nào cũng đầy ý tưởng, kể cả với một chương trình nhỏ như thế này, tôi không hề thiếu ý tưởng để cải tiến nó đâu ! 🤖

Chú ý: lần này tôi sẽ không đưa cho bạn đáp án nữa, bạn cần phải tự xoay sở lấy! Nếu bạn thật sự gặp vấn đề nan giải, hãy lên các diễn đàn có liên quan đến C. Tìm kiếm trước để xem thắc mắc của bạn đã có người giải thích chưa, nếu không thì tạo một topic khác để đặt câu hỏi. 🗣️

- Tạo một bộ đếm "số lần đoán". Bộ đếm này sẽ là một biến số tăng dần mỗi khi vòng lặp được lặp lại. VD: Khi người chơi tìm ra đáp án, sẽ thông báo "Chúc mừng ! Bạn đã tìm được số bí mật trong 8 lần đoán !!!"
- Khi người chơi tìm ra đáp án, chương trình dừng lại. Tại sao ta không hỏi người chơi có muốn chơi một ván khác không? Nếu bạn thực hiện điều này, chúng ta cần một vòng lặp bao lấy gần như toàn bộ chương trình, để người chơi không muốn dừng thì trò chơi vẫn được thực hiện lại. Tôi khuyên bạn hãy tạo một biến số dạng **Boolean** *tiếpTucGame* khởi tạo với giá trị 1. Nếu người chơi muốn dừng lại, hãy cho biến số giá trị 0 để chương trình dừng lại.
- Thêm vào mode chơi cho 2 người ! Cần chú ý là người chơi cần chọn giữa mode dành cho một người hay mode dành cho 2 người ! Bạn phải thêm vào một Menu ở đầu chương trình yêu cầu người chơi lựa chọn mode mong muốn. Sự khác nhau giữa 2 mode chơi là cách tạo *soBiMat*, trường hợp sử dụng *rand()* như ta vừa thấy, trường hợp sử dụng *scanf*.
- Tạo ra nhiều cấp độ khó khác nhau cho người chơi lựa chọn. Bắt đầu, tạo một Menu yêu cầu người chơi lựa chọn. Ví dụ:
 - 1 = từ 1 đến 100.
 - 2 = từ 1 đến 1000.
 - 3 = từ 1 đến 10000.

Nếu bạn thực hiện, bạn cần thay đổi *constant MAX*... Vâng! một *constant* thì không thể nào thay đổi giá trị suốt quá trình chạy chương trình! Đổi tên biến thành *soToiDa* (đồng thời bạn cần xóa đi từ khóa *const* nếu không giá trị đó vẫn là một *constant* đấy !). Giá trị của biến số này phụ thuộc vào cấp độ người chơi lựa chọn

Vậy đó, việc này sẽ khiến bạn phải mất thêm chút ít thời gian. 🌐

Hãy tận hưởng nó và đừng ngại thêm vào những ý tưởng khác để cải tiến thêm trò chơi của bạn. Và nếu như bạn có câu hỏi, đừng ngại lên các 4rum để thảo luận nhé 😊..
