BỘ GIAO THÔNG VÀ VẬN TẢI

TRƯỜNG ĐẠI HỌC GIAO THÔNG VÀ VẬN TẢI THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO**

**BÀI TẬP LỚN TÍNH ĐIỂM GIỮA KÌ**

**KĨ THUẬT LẬP TRÌNH**

|  |  |
| --- | --- |
| Giảng Viên | : Nguyễn Ngọc Thạch |
| Sinh viên thực hiện | : Trương Trần Thiên Bảo |
| Lớp | : DL22 |
| Khóa | : 22 |

**Thành phố Hồ Chí Minh, tháng 5 năm 2023**

**ĐỀ BÀI**

Cho hệ thức truy hồi:

a0 = 0, a1 =2

an = 5an-1 – 6an-2

Dùng C/C++ dể cài đặt tính số hạng thứ n bằng 5 cách:

1. Dùng hàm đệ quy
2. Dùng vòng lặp
3. Dùng mảng 1 chiều
4. Dùng mảng 2 chiều
5. Tìm nghiệm của hệ thức truy hồi rồi cài đặt

(Xây dựng 1 menu lựa chọn các cách)

Giải thích về các bước làm:

* Trong chương trình để xây dựng 1 menu lựa chọn các cách thì em đã khởi tạo 1 biến choice và cho nó bằng 0.
* Ngoài ra em còn khởi tạo hai biến a0 = 0, a1 = 2 đóng vai trò như là tham số trong các hàm ở ngoài hàm main() và khởi tạo 1 biến n để người dùng có thể nhập vào 1 số nguyên lớn hơn hoặc bằng 0.
* Sau đó em dùng vòng lặp do while để giúp cho việc in ra màn hình 5 cách để người dùng có thể nhập lựa chọn mà mình muốn sử dụng để tính toán số hạng thứ n.
* Em sẽ kiểm tra các điều kiện với số n được nhập vào từ bàn phím cũng như là lựa chọn của người dùng là nếu số nhập vào mà bé hơn 0 thì em sẽ có dòng lệnh in ra “Số bạn nhập không phù hợp” và ‘Vui lòng nhập lại số lớn hơn hoặc bằng 0”. Còn với lựa chọn thì em sẽ kiểm tra với điều kiện là nếu người dùng chọn cái lựa chọn mà không phải là các số trong đoạn từ 0 đến 5 thì sẽ yêu cầu nhập lại.
* Đối với 5 cách được thì e sẽ viết 5 hàm để có thể tính toán được số hạng thứ n. Ngoài ra em cũng viết thêm câu lệnh hàm giúp giải thích quá trình giải hệ thức truy hồi

1. Sử dụng hàm đệ quy.

* Đệ quy chính là một kĩ thuật lập trình trong đó một hàm gọi lại chính nó để giải quyết một vấn đề.
* Ở đây em đã viết một hàm với kiểu trả về là long long và em đặt tên hàm là solution\_1 nhận một tham số là int có tên là n và em cho hàm trả về giá trị long long

long long solution\_1(int n) {

return (n == 0) ? 0 : (n == 1) ? 2 : 5 \* solution\_1(n-1) – 6 \* solution\_1(n-2);

}

* Ở trong hàm solution\_1 này em đã kết hợp kĩ thuật đệ quy kết hợp với toán tử ba ngôi trong C++.
* Nếu nhập n bằng 2 vào thì kết quả của chương trình sẽ cho ra theo yêu cầu của đề bài là a2 = 5a1 – 6a0 tức là a2 = 5 \* 2 – 6 \* 0 tức là a2 = 10.
* Câu hỏi sẽ là câu lệnh trên sẽ được thực hiện như thế nào khi n = 2 ?
* Khi n = 2, câu lệnh kiểm tra dòng điều kiện (n == 0) thấy không thỏa điều kiện sẽ kiểm tra tiếp đến điều kiện tiếp theo là (n == 1) cũng không thỏa điều kiện nên sẽ thực hiện dòng lệnh 5 \* solution\_1(n-1) – 6 \* solution\_1(n-2) sẽ thành 5 \* solution\_1(2-1) – 6 \* solution\_1(2-2) vậy là hàm solution\_1 đã gọi lại chính nó với các tham số lần lượt là n-1 và n- 2, tức là 1 và 0.
* Như vậy ở solution\_1(n - 1) nhận đối số là n – 1 tức là 1 với n = 2 thì hàm kiểm tra điều kiện ( n == 0) không thỏa sẽ kiểm tra tiếp điều kiện ( n == 1) thấy thỏa điều kiện .Vậy hàm solution\_1(int n) với tham số int n = 1 sẽ trả kết quả là 2. (1)
* Còn hàm solution\_1(n – 2) nhận đối số n -2 tức là 0 với n = 2, hàm kiểm tra điều kiện đầu tiên thấy rằng ( n == 0) thỏa điều kiện. Vì vậy, hàm solution\_1(n-2) sẽ trả kết quả là 0. (2)
* Kết luận: từ (1) và (2) có thể thấy hàm solution đã thực hiện kĩ thuật đệ quy gọi lại chính nó với các đối số lần lượt là n -1 và n -2 có kết quả trả về lần lượt là 2 và 0. Vậy hàm sẽ được thực hiện với câu lệnh 5 \* solution\_1(n-1) – 6 \* solution\_1(n-2), tức là 5 \* 2 – 6 \* 0 = 10.
* Vậy hàm solution\_1 đã trả về đúng kết quả yêu cầu đề bài đặt ra.

1. Sử dụng vòng lặp

* Để tính được số hạng thứ n bằng vòng lặp bằng cách viết 1 hàm và đặt tên là solution\_2(int n) nhận tham số int và có tên là n, trong dó sử dụng vòng lặp for để tính toán số hạng thứ n.
* Hàm trả về là kiểu dữ liệu long long

long long solution\_2(int n) {

long long a0 = 0, a1 = 2, a =0;

for (long long i = 2; i <= n; ++i) {

a = 5 \* a1 – 6 \* a0;

a0 = a1; a1 =a;

}

Return a;

}

* Ví dụ người dùng nhập vào n là 3 thì câu lệnh sẽ được thực hiện như sau:

+ ở vòng lặp đầu i = 2, kiểm tra thấy i <= 3 , tăng i lên 1 đơn vị. Thực hiện tính toán a = 5 \* 2 – 6 \* 0 = 10, sau đó thực hiện gán giá trị a0 = a1 tức là a0 = 2 và tiếp theo là thực hiện gán a1 = a = 10 với kết quả vừa tính được của a

+ Mục đích ở đây chính là hoán đổi giá trị cho a1 và a0 để khi qua vòng lặp kế tiếp chúng ta có thể tính được số hạng thứ n bằng công thức a = 5 \* a1 – 6 \* a0 và việc hoán đổi giá trị không làm ảnh hưởng đển giá trị a0 và a1 ở ngoài vòng lặp.

+ i tăng lên 1 đơn vị tức là i = 3, kiểm tra thấy i <= 3 thực hiện câu lệnh trong vòng lặp. Với giá trị a1 và a0 đã được thay đổi trong vòng lặp thì a sẽ được tính toán như sau: a = 5 \* 10 – 6 \* 2 tức là a = 38

* Sau khi kết thúc vòng lặp, a = 38 thì return a; tức là trả về giá trị của a tức là 38
* Vậy là hàm solution\_2 sẽ trả kết quả là 38 đúng với yêu cầu cùa đề bài.

1. Dùng mảng 1 chiều

* Để có thể tính toán được số hạng thứ n bằng cách viết một hàm có tên là solution\_3 nhận tham số là int có tên là n.
* Hàm có kiểu trả về là long long.

long long solution\_3(int n) {

int \*a = new long long[n+1];

a[0] = 0, a[1] = 2;

for (long long int i = 2, i < long long(n+1); ++i) {

a[i] = 5 \* a[i-1] – 6 \* a[i-2];

}

return a[n];

delete[] a;

a = NULL;

}

* Trong hàm em đã cấp phát một mảng gồm n+1 phần tử để giúp có thể tính được số hạng thứ n vì chỉ số phần tử mảng bắt đầu từ 0 đến n – 1. Và con trỏ a sẽ giúp quản lí cái mảng này.
* Gán giá trị a[0] = 0, a[1] = 2
* Thực hiện vòng lặp với n = 2 tức là i = 2 thỏa điều kiện i < 3 nên thực hiện câu lệnh trong vòng lặp: a[2] = 5 \* a[1] – 6 \* a[0] = 5 \* 2 – 6 \* 0 = 10.
* Vòng lặp kết thúc sẽ đến câu lệnh return a[n] thì với n = 2 thì câu lệnh sẽ thành return a[2] tức là giá trị tại chỉ số thứ hai của mảng.
* Vậy câu lệnh sẽ trả kết quả là 10.
* Sau đó thực hiện giải phòng bộ nhớ cho mảng mà con trỏ a đang quản lí và gán giá trị NULL cho con trỏ a với câu lệnh a = NULL

1. Dùng mảng 2 chiều

* Em đã viết một hàm là solution\_4 nhận tham số là int có tên là n. Kiểu trả về là long long.
* Để dùng mảng 2 chiều tính toán được số hạng thứ n, sử dụng cấp phát động một mảng gồm 2 con trỏ kiểu long long\*. Mỗi con trỏ sẽ quản lí một hàng của mảng hai chiều.
* Sau đó sử dụng vòng lặp for để cấp phát động cho từng hàng của mảng 2 chiều. Mỗi hàng sẽ được cấp phát với câu lệnh a[i] = new int [n+1].

long long solution\_4(int n) {

long long \*\*a = new long long\*[2];

for (long long i = 0; i < 2; ++i) {

a[i] = new long long[n+1];

}

a[0][0] = 0; a[0][1] = 2;

for (long long j = 2; j < (long long)n+1; ++j) {

a[0][j] = 5 \* a[0][j-1] – 6 \* a[0][j-2];

}

return a[0][n];

for (long long i = 0; i < 2; ++i) {

delete[] a[i];

}

delete[] a;

a = NULL;

}

* Để tính toán số hạng thứ n thì quy ước a[0][n] chính là số hạng thứ n cần tìm
* Cho vòng lặp với chỉ số cột for (long long j = 2, j < (long long) n + 1; ++j)
* Với giá trị n nhập vào mà nếu j thỏa điều kiện j < n + 1 thì thực hiện câu lệnh trong vòng lặp.
* Ví dụ với n = 2 thì j = 2 sẽ bé hơn 3 thực hiện câu lệnh a[0][2] = 5 \* a[0][1] – 6 \* a[0][0] tức là a[0][2] = 5 \* 2 – 6 \* 0 = 10
* Sau đó return a[0][n] tức a[0][2], trả về kết quả 10.
* Tiếp theo dùng vòng lặp for để giải phóng bộ nhớ cho các hàng mà không sử dụng tới. Rồi delete[] a và gán a = NULL.

1. Tìm nghiệm của hệ thức truy hồi.

* Với đề bài đã cho:
* ta xét phương trình đặc trưng: r^2 – 5r + 6 = 0
* phương trình đặc trưng có 2 nghiệm r1 = 3, r2 = 2.
* Suy ra công thức nghiệm của hệ thức truy hổi đã cho có dạng: x \* 3^n + y \* 2^n.
* Xét hai phương trình:

a0 = x \* 3^n + y \* 2^n ⬄ x + y = 0 (1)

a1 = x \* 3^n + y \* 2^n ⬄3x +2y = 2 (2)

* Từ (1) và (2) suy ra x = 2, y = -2
* Từ đó công thức nghiệm của hệ thức truy hồi: 2 \* 3^n – 2 \* 2^n
* Từ công thức nghiệm truy hồi, viết hàm solution\_5 nhận các tham số int lần lượt là a0, a1, hai nghiệm của hệ phương trình đặc trưng là r1, r2 và n.
* Cho hệ số a, b tương đương x, y trong công thức nghiệm của hệ thức truy hồi.
* Tính hệ số b = (a0 – (2 \* a1)) / 6, a = (a0 – (6 \* b)) / 6.
* Sau đó kết hợp với hàm pow để trả về kết quả

long long solution\_5(int a0, int a1, int r1, int r2, int n) {

int b = (a0 – (2 \* a1)) / 6, a = (a0 – (6 \* b)) / 6;

return (a \* long long(pow(r1, n))) + (b \* long long(pow(r2, n)));

}

* Ví dụ nhập n = 2 thì hàm sẽ trả kết quả là 10.

Đây là bài báo cáo của em. Nếu có sai sót, mong thầy góp ý để e rút kinh nghiệm cho những bài báo cáo sau ạ.