

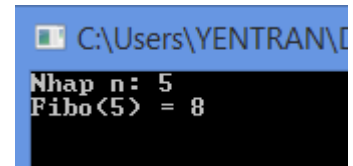
DANH SÁCH KÊ

(1) Sửa bài tập về nhà buổi 6:

a) *Khử đệ quy cho bài toán tính số hạng thứ n của dãy số Fibonacci được định nghĩa đệ quy như sau:*

$$F(0) = F(1) = 1$$

$$F(n) = F(n-1) + F(n-2) \text{ nếu } n \geq 2$$



```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

```
#include<conio.h>
```

```
#define MAXSIZE 50
```

```
//Hàm trả về giá trị số Fibo(n) sử dụng kỹ thuật đệ quy
```

```
int F(int n)
```

```
{
    if (n == 0 || n == 1)
        return 1;
    else
        return F(n - 1) + F(n-2);
}
```

```
/*Hàm trả về giá trị số Fibo(n) sử dụng kỹ thuật khử đệ quy:
```

```
Dùng mảng F để lưu giá trị các số Fibo(0), Fibo(1),..., Fibo(n)*/
```

```
int Fibo(int *F,int n)
```

```
{
    if (n == 0 || n == 1)
        return 1;//vì F[0] = F[1] = 1
    else
    {
        F[0] = F[1] = 1;
        for (int i = 2; i <= n; i++)
            F[i] = F[i - 1] + F[i - 2];
        return F[n];
    }
}
```

```
void main()
```

```
{
    int n;
    do
    {
        printf("Nhap n: ");
        scanf("%d",&n);
    } while (n<0);
    int *F;
    F=(int *)malloc((n+1)*sizeof(int));/*cấp phát vùng nhớ động cho mảng F gồm n+1 phần tử
                                         kiểu int: F[0],F[1],..., F[n] để lưu giá trị các số Fibo(0),
                                         Fibo(1),..., Fibo(n)*/
}
```

```

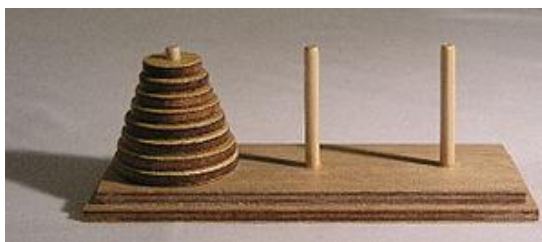
if (F != NULL) //Nếu cấp phát vùng nhớ động thành công
{
    printf("Fibo(%d) = %d", n, Fibo(F, n)); /*In giá trị số Fibo(n) được tính trong hàm
                                           Fibo(F,n)*/
    free(F); //giải phóng vùng nhớ động đã cấp phát cho mảng F
}
else
    printf("Loi cap phat vung nho dong");
_getch();
}

```

b) Sử dụng kỹ thuật đệ quy để viết chương trình minh họa trò chơi toán học “Tháp Hà Nội” (https://vi.wikipedia.org/wiki/Tháp_Hà_Nội)

Trò chơi này gồm một bộ các đĩa kích thước khác nhau, có lỗ ở giữa, nằm xuyên trên ba cái cọc. Bài toán đồ bắt đầu bằng cách sắp xếp các đĩa theo trật tự kích thước vào một cọc sao cho đĩa nhỏ nhất nằm trên cùng, tức là tạo ra một dạng hình nón. Yêu cầu của trò chơi là di chuyển toàn bộ số đĩa sang một cọc khác, tuân theo các quy tắc sau:

- ✓ Chỉ có 3 cột để di chuyển.
- ✓ Một lần chỉ được di chuyển một đĩa (không được di chuyển đĩa nằm giữa).
- ✓ Một đĩa chỉ có thể được đặt lên một đĩa lớn hơn (không nhất thiết hai đĩa này phải có kích thước liền kề, tức là đĩa nhỏ nhất có thể nằm trên đĩa lớn nhất).



Thuật giải đệ quy:

- ✓ Đặt tên các cọc là A, B, C -- những tên này có thể chuyển ở các bước khác nhau (ở đây: A = Cọc Nguồn, B = Cọc Đích, C = Cọc Trung Gian)
- ✓ Gọi n là tổng số đĩa.
- ✓ Đánh số đĩa từ 1 (nhỏ nhất, trên cùng) đến n (lớn nhất, dưới cùng)

Để chuyển n đĩa từ cọc A sang cọc B thì cần:

1. Chuyển n-1 đĩa từ A sang C. Chỉ còn lại đĩa #n trên cọc A
2. Chuyển đĩa #n từ A sang B
3. Chuyển n-1 đĩa từ C sang B cho chúng nằm trên đĩa #n

Phương pháp trên được gọi là thuật giải đệ quy: để tiến hành bước 1 và 3, áp dụng lại thuật giải cho n-1. Toàn bộ quá trình là một số hữu hạn các bước, vì đến một lúc nào đó thuật giải sẽ áp dụng cho n = 1. Bước này chỉ đơn giản là chuyển một đĩa duy nhất từ cọc A sang cọc B.

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
void Chuyen(int n, char cotnguồn, char cotdich, char cottrunggian)
{
    if (n == 1)
        printf("Chuyen dia %d tu cot nguon %c sang cot dich %c\n",1,cotnguồn,cotdich);
    else
    {
        Chuyen(n - 1, cotnguồn, cottrunggian, cotdich);
        printf("Chuyen dia %d tu cot nguon %c sang cot dich %c\n",n,cotnguồn,cotdich);
        Chuyen(n-1,cottrunggian,cotdich,cotnguồn);
    }
}
void main()
{
    int n;
    do
    {
        printf("Nhap n>0: ");
        scanf("%d",&n);
    } while (n<=0);
    Chuyen(n, 'A', 'B', 'C');
    _getch();
}

```

```

C:\Users\YENTRAN\Desktop\CTDL-T3H\Cons...
Nhap n>0: 3
Chuyen dia 1 tu cot nguon A sang cot dich B
Chuyen dia 2 tu cot nguon A sang cot dich C
Chuyen dia 1 tu cot nguon B sang cot dich C
Chuyen dia 3 tu cot nguon A sang cot dich B
Chuyen dia 1 tu cot nguon C sang cot dich A
Chuyen dia 2 tu cot nguon C sang cot dich B
Chuyen dia 1 tu cot nguon A sang cot dich B

```

```

C:\Users\YENTRAN\Desktop\CTDL-T3H\ConsoleAp..
Nhap n>0: 4
Chuyen dia 1 tu cot nguon A sang cot dich C
Chuyen dia 2 tu cot nguon A sang cot dich B
Chuyen dia 1 tu cot nguon C sang cot dich B
Chuyen dia 3 tu cot nguon A sang cot dich C
Chuyen dia 1 tu cot nguon B sang cot dich A
Chuyen dia 2 tu cot nguon B sang cot dich C
Chuyen dia 1 tu cot nguon A sang cot dich C
Chuyen dia 4 tu cot nguon A sang cot dich B
Chuyen dia 1 tu cot nguon C sang cot dich B
Chuyen dia 2 tu cot nguon C sang cot dich A
Chuyen dia 1 tu cot nguon B sang cot dich A
Chuyen dia 3 tu cot nguon C sang cot dich B
Chuyen dia 1 tu cot nguon A sang cot dich C
Chuyen dia 2 tu cot nguon A sang cot dich B
Chuyen dia 1 tu cot nguon C sang cot dich B

```

(2) Lý thuyết:

- Xem bài giảng CHƯƠNG 3 và giáo trình trang 35-48.

(3) Bài tập tại lớp:

- Viết các chương trình theo các ví dụ trong slide bài giảng chương 3:

Khai báo 1 cấu trúc danh sách kê gồm num nút kiểu int

```
#define MAX 50
struct LIST
{
    int num; //Số nút hiện có trong danh sách
    int nodes[MAX]; //Mỗi nút của danh sách là một phần tử trên mảng
};
```

Xây dựng CT có menu lệnh chọn lựa các xử lý danh sách kê như sau:

- Khởi động danh sách kê (num=0)
- Xác định số nút trong danh sách kê (num)
- Kiểm tra danh sách có bị rỗng hay không (num=0 => d/s rỗng)
- Kiểm tra danh sách có bị đầy hay không (num=MAX)
- Thêm một nút vào danh sách (xem lại giải thuật thêm 1 phần tử có giá trị gt vào mảng tại vị trí vt đã học trong môn KTLT C - Nội dung chính buổi 10)
- Xóa một nút khỏi danh sách (xem lại giải thuật xóa 1 phần tử của mảng tại vị trí vt đã học trong môn KTLT C – Nội dung chính buổi 10)
- Thay nút trong danh sách bằng một nút khác (cập nhật giá trị x khác cho nodes thứ pos: nodes[pos]=x)
- Sắp xếp danh sách theo thứ tự tăng dần / giảm dần (xem lại giải thuật sắp xếp mảng tăng dần / giảm đã học trong môn KTLT C - Nội dung chính buổi 10)
- Tìm kiếm một nút theo 2 phương pháp: tìm kiếm tuyến tính và tìm kiếm nhị phân. (xem lại 2 giải thuật tìm kiếm tuyến tính và tìm kiếm nhị phân đã học trong môn KTLT C - Nội dung chính buổi 10)
- Duyệt tất cả các nút của danh sách (kết xuất ra màn hình giá trị các nút trong d/s: hàm xuất mảng)
- Xoá tất cả các nút trong danh sách (num=0)

(4) Bài tập về nhà:

- Làm các bài tập có hướng dẫn 1, 2, 3, 4, 5 giáo trình trang 146-149.
- Viết chương trình có cài đặt danh sách kê để quản lý danh sách viên (có cấu trúc gồm: mã SV kiểu số, họ tên SV kiểu chuỗi). CT có các chức năng như sau:
 - ✓ Thêm 1 SV vào danh sách
 - ✓ Xóa 1 SV khỏi danh sách
 - ✓ Hiệu chỉnh SV
 - ✓ Xem danh sách SV
 - ✓ Sắp xếp danh sách theo mã SV
 - ✓ Tìm kiếm SV theo mã SV
 - ✓ Xóa toàn bộ danh sách

(Xem chương trình minh họa trong giáo trình trang 42-48)