

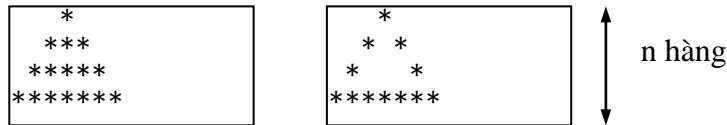
II. Bài Tập Thực Hành

Phần I. Ôn Tập ngôn ngữ C

- 1) Viết chương trình in bảng cửu chương từ 2 đến 9 ra màn hình.
- 2) Viết chương trình in tam giác vuông ra màn hình



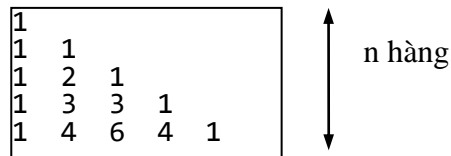
- 3) Viết chương trình in tam giác cân ra màn hình



- 4) Viết chương trình in chữ x ra màn hình



- 5) Cho tam giác Pascal được định nghĩa như sau:



Trong đó hàng thứ 0 có 1 phần tử $a[0][0] = 1$.

Hàng thứ i có $i+1$ phần tử:

$$\begin{aligned}a[i][0] &= 1 \\a[i][i] &= 1 \\a[i][j] &= a[i-1][j-1] + a[i-1][j]\end{aligned}$$

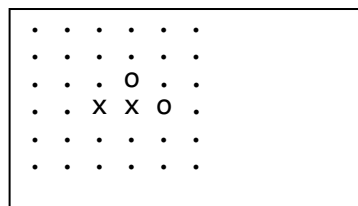
Viết chương trình nhập vào số n và in ra màn hình tam giác Pascal có chiều cao n .

- 6) Viết chương trình cuộn dòng chữ quảng cáo ngang màn hình.
- 7) Viết chương trình mô phỏng một trái banh di chuyển chéo trong màn hình. Trái banh đại diện bởi ký tự '*', có màu vàng, dịch chuyển trên nền xanh nước biển. Trái banh sẽ đổi lại, khi đụng đến cạnh màn hình.

Gợi ý:

- Dùng hàm `gotoxy()` để di chuyển con trỏ.
- Dùng hàm `textcolor()`, `textbackground()` để đặt màu, hàm `delay()` để giảm tốc độ di chuyển quả banh.

- 8) * Tương tự như bài 6 nhưng có bổ sung thêm hệ thống tường bằng các viên gạch, khi banh bay đụng gạch làm vỡ gạch (trò chơi Akarball).
- 9) * Viết chương trình chơi trò chơi caro 2 người



Gợi ý:

- Dùng hàm `getch()` để kiểm tra và đọc phím. Mã số một số phím cần thiết: ESC: 27; mũi tên \uparrow : 0, 72; mũi tên \leftarrow : 0, 75; mũi tên \rightarrow : 0, 77; mũi tên \downarrow : 0, 80; SPACEBAR: 32;
 - Dùng hàm `gotoxy()` để di chuyển con trỏ.
- 10) * Viết chương trình chơi trò chơi rắn săn mồi

Phần II. Đề Quy

- 1) Viết chương trình đệ quy tính giai thừa của số nguyên n ($n!$) nhập từ bàn phím. Debug chương trình với $n=4$
- 2) Viết chương trình đệ quy tìm số Fibonacci thứ n với n nhập từ bàn phím.

$$F_1 = F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

- 3) Cho tổ hợp: $C_n^k = \frac{n!}{k!(n-k)!}$ với $(0 < k < n)$

Người ta đã chứng minh tổ hợp trên cũng có thể tính bằng đệ quy qua công thức sau:

- $C_n^n = C_n^0 = 1$
- $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$

Hãy viết chương trình tính C_n^k theo cách tính đệ quy.

- 4) Viết chương trình phân tích số n thành tích các thừa số nguyên tố. Vd: $90 = 2 \cdot 3 \cdot 3 \cdot 5$.
- 5) Viết chương trình tìm USCLN của 2 số bằng đệ quy.
- 6) Viết chương trình đệ quy thực hiện bài toán tháp Hà Nội.
- 7) * Viết chương trình đệ quy thực hiện bài toán tháp Hà Nội, minh họa quá trình chuyển đĩa bằng hình vẽ.
- 8) Viết chương trình liệt kê tất cả các dãy gồm n bit nhị phân
- 9) Viết chương trình liệt kê các hoán vị của n số $1, 2, 3, \dots, n$
- 10) Cho tập A gồm n số $1, 2, 3, \dots, n$. Viết chương trình liệt kê các tập con k phần tử của A .
- 11) * Liệt kê các kết quả của bài toán 8 hậu dưới dạng bàn cờ.
- 12) * Cho vị trí của quân mã trên bàn cờ. Hãy tìm cách đi quân mã đến tất cả các ô trên bàn cờ.

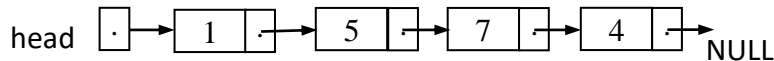
Phần III. Mảng và Sắp Xếp Mảng:

- 1) Nhập mảng n số nguyên. Đảo ngược mảng và in ra kết quả. Ví dụ: 3 5 4 7 → 7 4 5 3
- 2) Nhập mảng n số nguyên. Dịch phải mảng 1 vị trí (phần tử cuối mảng được chuyển về đầu). In ra kết quả. Ví dụ:
3 5 4 7 → 7 3 5 4
Debug chương trình với ví dụ trên.
- 3) Nhập mảng n số nguyên. Nhập vào số x, in ra vị trí xuất hiện của x trong mảng. Xóa số x trong mảng, in ra mảng sau khi xóa.
- 4) Nhập mảng gồm n số nguyên đã sắp xếp tăng dần. Nhập vào số x, dùng thuật toán BinarySearch in ra vị trí xuất hiện của x trong mảng.
- 5) Nhập mảng gồm n số nguyên đã sắp xếp tăng dần. Nhập vào số x. Chèn x vào mảng mà vẫn bảo đảm tính chất tăng dần của mảng.
- 6) Nhập số nguyên n. Dùng mảng, tính và in ra n số fibonacci đầu tiên.
- 7) Nhập mảng n số nguyên. Sắp xếp mảng theo thuật toán Selection Sort. In ra kết quả sắp xếp.
- 8) Thực hiện tương tự câu 7 cho các thuật toán:
 - a) Sắp xếp chèn (Insertion Sort)
 - b) Sắp xếp nổi bọt (Bubble Sort)
 - c) Sắp xếp nhanh (Quick Sort)
 - d) Sắp xếp trộn (Merge Sort)
- 9) Quản lý thông tin nhân viên trong một công ty gồm:
 - Mã nhân viên: chuỗi 8 ký tự
 - Tên nhân viên: chuỗi 20 ký tự
 - Lương: số nguyên
 - a. Tổ chức quản lý các nhân viên trong công ty bằng mảng
 - b. Lần lượt thêm các nhân viên vào mảng (n nhân viên).
 - c. In ra các nhân viên đã nhập.
 - d. Sắp xếp mảng theo thứ tự giảm dần của lương. In ra kết quả sắp xếp.
 - e. In ra tổng lương của các nhân viên.
 - f. Nhập vào một mã nhân viên x, cho biết tên và mức lương của nhân viên này. Sau đó xóa nhân viên này khỏi mảng.
- 10) * Viết chương trình minh họa chuyển động trên màn hình 3 thuật toán sắp xếp SelectionSort, Bubble Sort.
Gợi ý:
 - Mỗi số minh họa có 2 chữ số
 - Dùng hàm gotoxy() để in chuyển các chữ số.
- 11) * Viết chương trình minh họa chuyển động trên màn hình thuật toán sắp xếp InsertionSort.

Phần IV. Danh Sách Liên Kết:

- 1) Thực hiện danh sách liên kết đơn của các số nguyên:

Dùng hàm MakeNode, tạo ra danh sách liên kết 4 phần tử như sau:



In ra lại các phần tử theo thứ tự: nút cuối → nút kế cuối → nút kế đầu → nút đầu.

2) Thực hiện danh sách liên kết đơn của các số nguyên:

- Xây dựng các hàm khởi tạo, chèn đầu, chèn cuối.
- Hàm duyệt danh sách liên kết

Nhập vào n số nguyên, chèn cuối vào danh sách liên kết. Dùng hàm duyệt in ra lại kết quả.

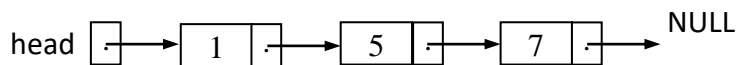
3) Nhập vào danh sách liên kết gồm n số nguyên (câu 2). Viết hàm `int Get(int i)` để truy xuất giá trị nút thứ i của danh sách.

4) Nhập vào danh sách liên kết gồm n số nguyên

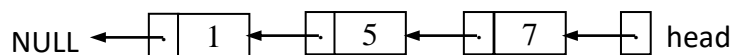
- Viết hàm đếm số phần tử có trong danh sách.
- Viết hàm tính tổng và hàm tính trung bình cộng các số trong danh sách
- Viết hàm kiểm tra xem các số trong danh sách có sắp theo thứ tự tăng dần hay không.

5) * Nhập vào danh sách liên kết gồm n số nguyên. Viết hàm đảo ngược một danh sách liên kết đơn cho trước bằng cách chỉ gán lại giá trị cho con trỏ next.

ví dụ: cho danh sách ban đầu:



đổi thành:



6) Quản lý thông tin sinh viên của một lớp gồm:

- Mã sinh viên: chuỗi 8 ký tự
- Tên sinh viên: chuỗi 20 ký tự
- Điểm Toán, Lý, Hóa: số thực

- Tổ chức quản lý các sinh viên trong lớp bằng danh sách liên kết đơn.
- Thêm các sinh viên vào cuối danh sách (n sinh viên).
- In ra những sinh viên có $DTB \geq 5$ với $DTB = (Toán + Lý + Hóa)/3$
- Nhập vào mã sinh viên. In ra thông tin về tên, điểm của SV. Sau đó xóa sinh viên khỏi danh sách.

7) Sửa lại trò chơi rắn săn mồi dùng danh sách liên kết.

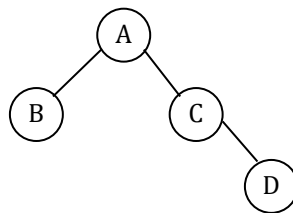
8) Trong bài toán Josephus, một nhóm binh sĩ bị kẻ thù bao vây và một binh sĩ được chọn đi cầu cứu. Việc chọn thực hiện theo cách sau: chọn một số nguyên n và các binh sĩ được sắp thành vòng tròn. Bắt đầu từ một binh sĩ ngẫu nhiên đếm theo chiều kim đồng hồ. Khi đếm đến n, binh sĩ đó ra khỏi vòng và binh sĩ tiếp theo bắt đầu đếm lại từ 1. Quá trình này diễn ra cho tới khi chỉ còn lại một binh sĩ là binh sĩ được chọn. Dùng danh sách liên kết vòng, viết chương trình cài đặt cách chọn này.

Phần IV. Stack và Queue

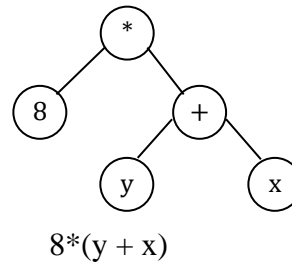
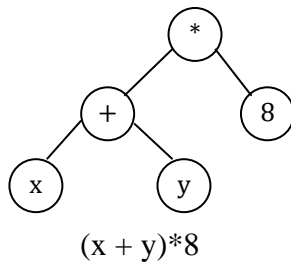
- 1) Viết chương trình dùng Stack để chuyển một số thập phân sang nhị phân
- 2) Viết chương trình phân tích một số nguyên n ra thừa số nguyên tố. In các thừa số theo thứ tự từ lớn đến nhỏ. Ví dụ $n = 3960$, chương trình hiển thị: $3960 = 11 * 5 * 3 * 3 * 2 * 2 * 2$
- 3) Viết chương trình dùng Stack tính một biểu thức dạng hậu tố. (Biểu thức là một chuỗi chỉ gồm các con số nguyên và các phép toán $+$, $-$, $*$, $/$, $^$, các phần tử cách nhau 1 khoảng trắng. Ví dụ: “12 5 + 8 -“)
- 4) * Viết chương trình dùng Stack để đổi một biểu thức từ dạng trung tố sang dạng hậu tố. (Biểu thức là một chuỗi chỉ gồm các con số nguyên và các phép toán $+$, $-$, $*$, $/$, $^$, các phần tử cách nhau 1 khoảng trắng)
- 5) * Từ bài 4 viết chương trình vẽ đồ thị của một hàm số nhập từ bàn phím.

Phần V. Cây Nhị Phân

- 1) Dùng hàm MakeNode, tạo ra cây nhị phân sau (Thuộc tính info là kiểu chuỗi 20 ký tự):



- a) In ra lại các phần tử theo thứ tự: nút gốc \rightarrow nút con trái của nút gốc \rightarrow nút con phải nút gốc \rightarrow nút con phải của nút con phải.
 - b) Xóa cây.
- 2) Thực hiện lại câu 1, bổ sung các hàm sau: duyệt theo thứ tự giữa, duyệt theo thứ tự sau.
 - 3) Thực hiện lại câu 1, bổ sung các hàm sau:.
 - a) Đếm số nút trong cây
 - b) Đếm số nút lá trong cây
 - c) Tính chiều cao của cây.
 - 4) * Nhập vào một biểu thức dạng hậu tố (xem bài 3 phần IV). Viết chương trình chuyển biểu thức thành cây nhị phân (thuộc tính info của nút là kiểu chuỗi). Áp dụng phép duyệt theo thứ tự sau và duyệt theo thứ tự giữa để in ra lại biểu thức hậu tố và biểu thức trung tố tương ứng.
Gợi ý: để chuyển biểu thức hậu tố thành cây nhị phân: dùng một stack con trỏ thay cho stack số nguyên.
 - 5) Dựa vào bài 4, thực hiện các chức năng sau:
 - a) Đếm số nút trong cây (số toán tử + toán hạng)
 - b) Đếm số nút lá trong cây (số toán hạng)
 - c) Tính chiều cao của cây.
 - d) Đảo ngược cây, in ra lại biểu thức trung tố.



- 6) Cho hai cây nhị phân, viết chương trình kiểm tra xem hai cây có cùng cấu trúc với nhau hay không (đẳng cấu).
- 7) Nhập mảng n phần tử. In ra kết quả sắp xếp theo thuật toán Heap Sort.

Phần VII. Cấu trúc tìm kiếm

- 1) Viết chương trình sau trên cây nhị phân tìm kiếm
 - Lần lượt thêm các số vào cây (n số).
 - In kết quả duyệt theo thứ tự giữa ra màn hình (kết quả phải là dãy tăng)
 - Cho biết một số x có tồn tại trong cây hay không.
 - Xóa một số x khỏi cây nhị phân tìm kiếm. In ra lại kết quả để kiểm tra.
- 2) Thực hiện bảng băm m số nguyên giải quyết đụng độ bằng phép thử tuyến tính:
 - a. Nhập vào một số x. Bổ sung x vào bảng băm
 - b. Nhập vào số x. In ra vị trí của x trong bảng băm.