

Lời nói đầu

250 bài tập kỹ thuật lập trình C (230 bài tập chính thức, 20 bài tập bổ sung) trong tập sách này được chọn lọc từ các bài tập thực hành môn Ngôn ngữ lập trình C và Lập trình Cấu trúc dữ liệu bằng ngôn ngữ C cho sinh viên Đại học và Cao đẳng chuyên ngành Công nghệ Thông tin.

Các bài tập đã được sắp xếp theo một trình tự nhất định, nhằm đảm bảo cho người đọc nắm vững một cách có hệ thống các kiến thức cần thiết của kỹ thuật lập trình nói chung và ngôn ngữ lập trình C nói riêng; chuẩn bị nền tảng cho các môn học có liên quan. Mặc dù cố gắng duyệt qua các vấn đề cơ bản của ngôn ngữ lập trình C, nhưng tập sách này được viết với mục tiêu củng cố và nâng cao khả năng làm việc với ngôn ngữ C.

Khác với các sách bài tập khác, các bài tập trong tập sách này đều có hướng dẫn giải chi tiết. Khi hướng dẫn giải bài tập, chúng tôi cố gắng:

- Thể hiện một góc nhìn riêng về kỹ thuật lập trình bằng ngôn ngữ C, chú ý đến những đặc điểm của ngôn ngữ C. Nói cách khác, chúng tôi chú ý đến lập trình theo phong cách của C.
- Phân tích quá trình tư duy khi giải quyết vấn đề, củng cố các kiến thức toán học cũng như lập trình cơ bản, nhằm làm nổi bật vai trò của ngôn ngữ lập trình như một công cụ hỗ trợ mang tính thực tế cao.
- Lập trình thật ngắn gọn và rõ ràng giúp người đọc hiểu rõ vấn đề. Nâng cao kỹ năng lập trình. Người đọc sẽ thấy thú vị và bất ngờ với một số kỹ thuật giải quyết vấn đề.
- Theo chuẩn ANSI/ISO C89 phù hợp với nhà trường ở Việt nam, chuẩn mới nhất là ANSI/ISO C11 (ISO/IEC 9899:2011).
- Các bài giải của 250 bài tập và các phương án giải khác đã được kiểm tra bằng Cppcheck 1.72 (cppcheck.sourceforge.net).

Chúng tôi tin rằng tập sách này sẽ giúp người đọc thật sự củng cố và nâng cao kiến thức lập trình với ngôn ngữ C.

Mặc dù đã dành rất nhiều thời gian và công sức cho tập sách, phải hiệu chỉnh nhiều lần và chi tiết, nhưng tập sách không thể nào tránh được những sai sót và hạn chế. Chúng tôi thật sự mong nhận được các ý kiến góp ý từ bạn đọc để tập sách có thể hoàn thiện hơn.

Xin chân thành cảm ơn anh Lê Gia Minh đã xem và đóng góp nhiều ý kiến quý giá cho tập sách. Cảm ơn bạn Nguyễn Đình Song Toàn đã khuyến khích tôi học C. Cảm ơn các anh Thân Văn Sử, Lê Mậu Long, Nguyễn Minh Nam, tôi đã học tập được rất nhiều kinh nghiệm từ các anh.

Phiên bản

Cập nhật ngày: 08/03/2016

Thông tin liên lạc

Mọi ý kiến và câu hỏi có liên quan xin vui lòng gửi về:

Dương Thiên Tứ

91/29 Trần Tấn, P. Tân Sơn Nhì, Q. Tân Phú, Thành phố Hồ Chí Minh

Facebook: <https://www.facebook.com/tu.duongthien>

E-mail: thientu2000@yahoo.com

Hướng dẫn sử dụng tài liệu

Trong giáo trình thực hành này, các bạn sẽ thực hiện các bài tập lập trình cơ bản, được thực hiện bằng ngôn ngữ lập trình C, theo chuẩn ANSI/ISO C89 (ANS X3.159-1989 và ISO/IEC 9899 - 1990).

ANSI/ISO C99 (ISO/IEC 9899 - 1999) hiện chưa dùng phổ biến tại nhà trường ở Việt nam, bạn có thể tham khảo thêm từ các tài liệu giới thiệu trong phần tham khảo.

Hướng dẫn thực hiện bài tập thực hành

- Các bạn nên thực hiện toàn bộ các bài tập thực hành. Các bài tập này đã được tuyển chọn và sắp xếp để mang đến cho các bạn kiến thức cơ bản và tổng quát về ngôn ngữ lập trình C. Các bạn nên:

✎ Đọc kỹ bài tập để hiểu rõ yêu cầu bài tập.

✎ Dành nhiều thời gian thiết kế cẩn thận chương trình. Nhiều vấn đề lập trình sẽ nảy sinh do thiết kế sai, và nếu bạn mất nhiều thời gian để thiết kế bạn sẽ rút ngắn được giai đoạn viết code và dò lỗi. Luôn luôn thử tìm một cách đơn giản nhất để thiết kế chương trình.

- Nếu chương trình có lỗi và không chạy được, trước khi xem bài giải, hãy chắc rằng bạn đã:

✎ Mất nhiều thời gian để cố gắng giải bài tập theo cách của bạn;

✎ Thử dùng tiện ích dò lỗi (debugger) nếu chương trình có lỗi;

✎ Đọc kỹ lại bài học lý thuyết có liên quan;

✎ Thử mọi cách mà bạn nghĩ có thể giải được bài tập.

- Một số chi tiết:

✎ Các chương trình không yêu cầu kiểm tra chặt chẽ dữ liệu nhập. Tuy nhiên, có thể dùng hàm `assert()` để kiểm tra các tiền điều kiện (pre-condition).

✎ Các bài tập có thể thực hiện hai phiên bản: giải quyết vấn đề trực tiếp trong hàm `main()`, hoặc viết các hàm phụ để giải quyết từng vấn đề riêng tùy theo yêu cầu và độ phức tạp của bài tập (hàm `main()` xem như một test driver).

✎ Các bài tập về mảng (array) và chuỗi (string) thực hiện hai phiên bản: không dùng con trỏ và dùng con trỏ (cấp phát động).

- Xem bài giải:

Bài giải chỉ trình bày một trong các lời giải có thể có của bài tập. Chúng tôi đã cố đa dạng hóa cách giải để bạn có thể rút được nhiều kiến thức và kinh nghiệm từ bài giải. Bạn cũng có thể học tập thêm cách tiếp cận vấn đề, cách viết code, ...

Bạn chỉ xem bài giải khi đã thực hiện xong bài tập, so sánh với bài giải của bạn để có thêm kinh nghiệm.

Ghi chú dùng trong sách



Thông tin, kiến thức hỗ trợ cần có để thực hiện bài tập.



Ví dụ xuất mẫu của chương trình.
Dùng để kiểm tra nhanh chương trình.



Gợi ý giải bài tập.

KHÁI NIỆM CƠ BẢN - TOÁN TỬ

CẤU TRÚC LỰA CHỌN - CẤU TRÚC LẬP

Bài 1: Nhập vào diện tích S của một mặt cầu. Tính thể tích V của hình cầu này.



$$\begin{cases} S = 4\pi R^2 \\ V = \frac{4}{3}\pi R^3 \end{cases} \quad (\pi \approx 3.141593)$$



Nhap dien tich S: 256.128 ↵
The tich V = 385.442302

Bài giải: xem trang 66

Bài 2: Nhập vào tọa độ 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$. Tính khoảng cách AB .



$$|AB| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$



A(x_A, y_A)? 3.2 -1.4 ↵
B(x_B, y_B)? -5.7 6.1 ↵
|AB| = 11.6387

Bài giải: xem trang 66

Bài 3: Viết chương trình nhập vào tọa độ (x_C, y_C) là tâm của một đường tròn, và R là bán kính của đường tròn đó. Nhập vào tọa độ (x_M, y_M) của điểm M . Xác định điểm M nằm trong, trên hay nằm ngoài đường tròn.



Nhap toa do tam C(x_C, y_C)? 0.5 4.3 ↵
Nhap ban kinh R? 7.4 ↵
Nhap toa do M(x_M, y_M)? 3.2 6.5 ↵
M nam trong C()

Bài giải: xem trang 66

Bài 4: Viết chương trình nhập vào ba số thực là ba cạnh của một tam giác. Kiểm tra ba cạnh được nhập có hợp lệ hay không. Nếu hợp lệ, hãy cho biết loại tam giác và tính diện tích tam giác đó.



Tổng hai cạnh bất kỳ của một tam giác phải lớn hơn cạnh còn lại.
Công thức Heron¹ dùng tính diện tích tam giác theo chu vi:

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \text{ trong đó } p \text{ là nửa chu vi: } p = \frac{a+b+c}{2}$$



Nhap 3 canh tam giac: 3 4 5 ↵
Tam giac vuong
Dien tich S = 6

Bài giải: xem trang 67

Bài 5: Viết chương trình nhập vào tọa độ các đỉnh của tam giác ABC và của điểm M . xác định điểm M nằm trong, nằm trên cạnh hay nằm ngoài tam giác ABC .

¹ Heron of Alexandria (10 - 70)



Công thức tính diện tích một tam giác theo tọa độ 3 đỉnh của nó:

$$S_{ABC} = \frac{1}{2} \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix} = \frac{1}{2} \left(x_A \begin{vmatrix} y_B & 1 \\ y_C & 1 \end{vmatrix} - y_A \begin{vmatrix} x_B & 1 \\ x_C & 1 \end{vmatrix} + \begin{vmatrix} x_B & y_B \\ x_C & y_C \end{vmatrix} \right)$$

$$= \frac{1}{2} |x_A(y_B - y_C) - y_A(x_B - x_C) + (x_B y_C - x_C y_B)|$$

$$= \frac{1}{2} |x_A y_B - x_B y_A + x_B y_C - x_C y_B + x_C y_A - x_A y_C|$$



Biện luận bằng cách so sánh tổng diện tích: $\Delta MAB + \Delta MBC + \Delta MCA$ với diện tích ΔABC .



```
A(xA, yA)? 0 5 ↵
B(xB, yB)? 3 0 ↵
C(xC, yC)? 4 7 ↵
M(xM, yM)? 2 6 ↵
M nằm trên cạnh tam giác ABC
```

Bài giải: xem trang 68

Bài 6: Viết chương trình nhập vào ba số nguyên. Hãy in ba số này ra màn hình theo thứ tự tăng dần và chỉ dùng tối đa một biến phụ.



```
Nhap a, b, c: 5 3 4 ↵
Tang dan: 3 4 5
```

Bài giải: xem trang 69

Bài 7: Viết chương trình giải phương trình bậc 1: $ax + b = 0$ (a, b nhập từ bàn phím). Xét tất cả các trường hợp có thể.



```
Nhap a, b: 4 -3
x = 0.75
```

Bài giải: xem trang 70

Bài 8: Viết chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$ (a, b, c nhập từ bàn phím). Xét tất cả các trường hợp có thể.



Nghiệm của phương trình bậc 2: $ax^2 + bx + c = 0$ ($a \neq 0$)

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}, \text{ với } \Delta = b^2 - 4ac$$



```
Nhap a, b, c: 2 1 -4 ↵
x1 = -6.74456
x2 = 4.74456
```

Bài giải: xem trang 71

Bài 9: Viết chương trình nhập vào số x chỉ số đo của một góc, tính bằng phút. Cho biết nó thuộc góc vuông thứ bao nhiêu của vòng tròn lượng giác. Tính $\cos(x)$, dùng hàm `math.h` cung cấp.



$60' = 1^\circ$. Công thức chuyển đổi giữa độ và radian: $1 \text{ radian} = \frac{180}{\pi} \text{ degree}$



Nhap so do x cua goc (phut): **12345** ↵
 x thuoc goc vuong thu 3
 $\cos(x) = -0.900698$

Bài giải: xem trang 72

Bài 10: Số bảo hiểm xã hội của Canada (SIN - Canadian Social Insurance Number) là một số có 9 chữ số, được kiểm tra tính hợp lệ như sau:

- Số phải nhất (vị trí 1, tính từ phải sang), là số kiểm tra (check digit).
 - Trọng số được tính từ phải qua trái (không tính check digit), bằng $s_1 + s_2$:
 + s_1 là tổng các số có vị trí lẻ.
 + Các số có vị trí chẵn nhân đôi. Nếu kết quả nhân đôi có hai chữ số thì kết quả là tổng của hai chữ số này. s_2 là tổng các kết quả.
- SIN hợp lệ có tổng trọng số với số kiểm tra chia hết cho 10.

Ví dụ: SIN 193456787

- Số kiểm tra là 7 (số tô đậm).
- Trọng số là tổng của s_1 và s_2 , với:

$$+ s_1 = 1 + 3 + 5 + 7 = 16$$

+ Các số có vị trí chẵn nhân đôi:

$$(9 * 2) (4 * 2) (6 * 2) (8 * 2) \Rightarrow 18 \ 8 \ 12 \ 16$$

$$s_2 = (1 + 8) + 8 + (1 + 2) + (1 + 6) = 27$$

$$\text{Trọng số bằng } s_1 + s_2 = 16 + 27 = 43.$$

Vì tổng trọng số với số kiểm tra $43 + 7 = 50$ chia hết cho 10 nên số SIN hợp lệ.

Viết chương trình nhập một số SIN. Kiểm tra xem số SIN đó có hợp lệ hay không.

Nhập 0 để thoát.



SIN (0 để thoát): **193456787** ↵
 SIN hop le!
 SIN (0 để thoát): **193456788** ↵
 SIN khong hop le!
 SIN (0 để thoát): **0** ↵

Bài giải: xem trang 72

Bài 11: Viết trò chơi bao - đá - kéo với luật chơi: bao thắng đá, đá thắng kéo, kéo thắng bao. Người dùng nhập vào một trong ba ký tự b (bao), d (đá), k (kéo); máy tính sinh ngẫu nhiên một trong ba ký tự trên, thông báo kết quả chơi.



Nhap ky tu (b-d-k), ky tu khac để thoát: **b** ↵
 Computer: d
 Ty so human - computer: 1 - 0
 Nhap ky tu (b-d-k), ky tu khac để thoát: **k** ↵
 Computer: d
 Ty so human - computer: 1 - 1
 Nhap ky tu (b-d-k), ky tu khac để thoát: **0** ↵

Bài giải: xem trang 73

Bài 12: Viết chương trình giải hệ phương trình 2 ẩn:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

Các hệ số $a_1, a_2, b_1, b_2, c_1, c_2$ nhập từ bàn phím. Xét tất cả các trường hợp cụ thể.



Công thức Cramer² dùng tính hệ phương trình 2 ẩn:

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \quad D_x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} \quad D_y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}$$

Nếu $D \neq 0$, $x = \frac{D_x}{D}$, $y = \frac{D_y}{D}$



Nhap a1, b1, c1: 1 2 3 ↵

Nhap a2, b2, c2: 4 5 6 ↵

x = -1

y = 2

Bài giải: xem trang 74

Bài 13: Viết chương trình nhập vào ngày, tháng, năm. Kiểm tra ngày và tháng nhập có hợp lệ hay không. Tính thứ trong tuần của ngày đó.



Năm nhuận (leap year) tính theo lịch Gregorian (từ 1582): năm phải chia hết cho 4 và không chia hết cho 100, hoặc năm phải chia hết cho 400.

Thứ trong tuần tính theo công thức Zeller³:

$$\text{dayofweek} = (d + y + y/4 - y/100 + y/400 + (31 * m)/12) \% 7$$

với: $a = (14 - \text{month})/12$

$y = \text{year} - a$

$m = \text{month} + 12*a - 2$

dayofweek: 0 (chúa nhật), 1 (thứ hai), 2 (thứ ba), ...



Nhap ngay, thang va nam: 20 4 1976 ↵

Hop le

Thu 3

Bài giải: xem trang 75

Bài 14: Viết chương trình nhập vào ngày, tháng, năm (giả sử nhập đúng, không cần kiểm tra hợp lệ). Tìm ngày, tháng, năm của ngày tiếp theo.

Tương tự, tìm ngày, tháng, năm của ngày trước đó.



Nhap ngay, thang, nam: 28 2 2000 ↵

Ngày mai: 29/02/2000

Nhap ngay, thang, nam: 1 1 2001 ↵

Hôm qua: 31/12/2000

Bài giải: xem trang 76

Bài 15: Viết chương trình nhập vào ngày, tháng, năm (giả sử nhập đúng, không cần kiểm tra hợp lệ). Tìm xem ngày đó là ngày thứ bao nhiêu trong năm.



Nếu không dùng vòng lặp, có thể dùng công thức sau:

$$\text{sum} = (\text{int}) (30.42 * (\text{month} - 1)) +$$



² Gabriel Cramer (1704 - 1752)

³ Julius Christian Johannes Zeller (1824 - 1899)

Nếu month = 2, hoặc năm nhuận và month > 2 thì sum = sum + 1
 Nếu 2 < month < 8 thì sum = sum - 1



Nhap ngay, thang, nam: 4 4 2000 ↵
 Ngay thu: 95

Bài giải: xem trang 77

Bài 16: Viết chương trình nhập vào một năm (> 1582), in lịch của năm đó. Tính thứ cho ngày đầu năm bằng công thức Zeller (bài 14, trang 6).



Nhap nam: 2008 ↵
 Thang 1

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

 ...
 Thang 12

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Bài giải: xem trang 78

Bài 17: Viết chương trình tạo lịch trực cho 5 bạn: A, B, C, D, E. Nhập năm và thứ (0 - 6, 0 là Chúa Nhật, 1 là thứ Hai, ...) cho ngày đầu năm. Sau đó nhập một tháng trong năm và in lịch trực của tháng đó. Lưu ý 5 bạn trực lần lượt theo thứ tự trên, ngày Chúa nhật không trực và bạn A sẽ trực ngày đầu tiên của năm.



Nhap nam: 2006 ↵
 Nhap thu cho ngay dau tien cua nam: 0 ↵
 Nhap thang: 5 ↵

Sun		Mon		Tue		Wen		Thu		Fri		Sat	
		1	[C]	2	[D]	3	[E]	4	[A]	5	[B]	6	[C]
7	[]	8	[D]	9	[E]	10	[A]	11	[B]	12	[C]	13	[D]
14	[]	15	[E]	16	[A]	17	[B]	18	[C]	19	[D]	20	[E]
21	[]	22	[A]	23	[B]	24	[C]	25	[D]	26	[E]	27	[A]
28	[]	29	[B]	30	[C]	31	[D]						

Bài giải: xem trang 80

Bài 18: Viết chương trình nhập vào số giờ, xuất ra số tương đương tính theo tuần, theo ngày và theo giờ.



Nhap so gio: 1000 ↵
 5 tuan, 6 ngay, 16 gio

Bài giải: xem trang 81

Bài 19: Nhập vào thời điểm 1 và thời điểm 2. Tìm thời gian trải qua giữa hai thời điểm này tính bằng giờ, phút, giây.



```
Nhap gio, phut, giây [1]: 3 28 47 ↵
Nhap gio, phut, giây [2]: 5 40 12 ↵
Hieu thời gian: 2 gio 11 phut, 25 giây
```

Bài giải: xem trang 81

Bài 20: Viết chương trình nhập số kW điện đã tiêu thụ. Tính tiền điện phải trả, biết rằng khung giá điện như sau:

0kW	100kW	250kW	350kW
500đ/kW	800đ/kW	1000đ/kW	1500đ/kW



```
Nhap so kW tiêu thụ: 4321 ↵
Chi phí: 6226500
```

Bài giải: xem trang 82

Bài 21: Trong kỳ thi tuyển, một thí sinh sẽ trúng tuyển nếu có điểm tổng kết lớn hơn hoặc bằng điểm chuẩn và không có môn nào điểm 0.

- Điểm tổng kết là tổng điểm của 3 môn thi và điểm ưu tiên.
- Điểm ưu tiên bao gồm điểm ưu tiên theo khu vực và điểm ưu tiên theo đối tượng.

Khu vực			Đối tượng		
A	B	C	1	2	3
2	1	0.5	2.5	1.5	1

Viết chương trình nhập: điểm chuẩn của hội đồng, điểm 3 môn thi của thí sinh, khu vực (nhập X nếu không thuộc khu vực ưu tiên) và đối tượng dự thi (nhập 0 nếu không thuộc đối tượng ưu tiên). Cho biết thí sinh đó đậu hay rớt và tổng số điểm đạt được.



```
Nhap diem chuan: 15.5 ↵
Nhap diem 3 mon thi: 4.5 3.4 3.6 ↵
Nhap khu vuc (A, B, C, X): B ↵
Nhap doi tuong (1, 2, 3, 0): 1 ↵
Rot [15]
```

Bài giải: xem trang 83

Bài 22: Viết chương trình liệt kê, đếm và tính tổng các ước số của số nguyên dương n (n nhập từ bàn phím).



```
Nhap n: 1966 ↵
Cac uoc so: 1 2 983 1966
Co 4 uoc so, tong la: 2952
```

Bài giải: xem trang 83

Bài 23: Viết chương trình tìm các số hoàn hảo (perfect number) nhỏ hơn một số nguyên dương n cho trước. Biết số hoàn hảo là số nguyên dương, bằng tổng các *ước số thực sự* của nó (ví dụ: $28 = 14 + 7 + 4 + 2 + 1$).



```
Nhap n: 10000 ↵
Cac so hoan hao nho hon 10000: 6 28 496 8128
```

Bài giải: xem trang 84

Bài 24: Nhập vào một số tự nhiên n (n khai báo kiểu unsigned long)

- Số tự nhiên n có bao nhiêu chữ số.
- Hãy tìm chữ số cuối cùng của n .
- Hãy tìm chữ số đầu tiên của n .
- Tính tổng các chữ số của n .
- Hãy tìm số đảo ngược của n .



```
Nhap n: 43210 ↵
43210 co 5 chu so
Chu so cuoi cung la: 0
Chu so dau tien la: 4
Tong cac chu so la: 10
So dao nguoc la: 1234
```

Bài giải: xem trang 84

Bài 25: Nhập vào hai số nguyên dương a, b . Tính ước số chung lớn nhất và bội số chung nhỏ nhất của a, b .



USCLN: (Greatest Common Divisor) $\gcd(a, b) = \max\{k \mid k \mid a \wedge k \mid b\}$
 BSCNN: (Least Common Multiple) $\text{lcm}(a, b) = \min\{k \mid k > 0, a \mid k \wedge b \mid k\}$



USCLN(a, b): + Cho gcd bằng a hoặc b
 + Trừ dần gcd cho đến khi cả a và b đều chia hết cho gcd
 + USCLN (a, b) = gcd
 BSCNN(a, b): + Cho lcm bằng a hoặc b
 + Tăng dần lcm cho đến khi lcm chia hết cho cả a và b
 + BSCNN (a, b) = lcm



```
Nhap cap (a, b): 12 8 ↵
USCLN (a, b): 4
BSCNN (a, b): 24
```

Bài giải: xem trang 86

Bài 26: Nhập vào tử số, mẫu số (đều khác 0) của một phân số. Hãy rút gọn phân số này. Chọn dạng xuất thích hợp trong trường hợp mẫu số bằng 1 và phân số có dấu.



Đề rút gọn một phân số, chia cả tử số và mẫu số cho USCLN của tử số và mẫu số.



```
Nhap tu so, mau so: -3 -15 ↵
Rut gon: 1/5
Nhap tu so, mau so: 8 -2 ↵
Rut gon: -4
```

Bài giải: xem trang 88

Bài 27: Nhập vào một số nguyên dương n , phân tích n thành các thừa số nguyên tố.



```
Nhap n: 12345 ↵
3 * 5 * 823
```

Bài giải: xem trang 89

Bài 28: Viết chương trình mô phỏng hàm ROUND của Microsoft Excel, dùng làm tròn một số double với một số n cho trước.



- Nếu $n > 0$, số làm tròn sẽ có n chữ số phần thập phân.
- Nếu $n = 0$, số làm tròn sẽ là số nguyên gần nhất.
- Nếu $n < 0$, số làm tròn là số nguyên làm tròn từ vị trí thứ n tính từ phải sang.



```
Nhap so thuc x: 3.1415926535 ↵
Do chinh xac: 7 ↵
3.1415927
Nhap so thuc x: -4.932 ↵
Do chinh xac: 0 ↵
-5
Nhap so thuc x: 21.5 ↵
Do chinh xac: -1 ↵
20
```

Bài giải: xem trang 90

Bài 29: Lập bảng so sánh hai thang đo nhiệt độ Fahrenheit và Celsius⁴ trong:

- Đoạn $[0^\circ\text{C}, 10^\circ\text{C}]$, bước tăng 1°C .
- Đoạn $[32^\circ\text{F}, 42^\circ\text{F}]$, bước tăng 1°F .



Công thức chuyển đổi Fahrenheit - Celcius:
 $5(F - 32) = 9C$



Celcius	Fahrenheit	Fahrenheit	Celcius
0	32.00	32	0.00
1	33.80	33	0.56
2	35.60	34	1.11
3	37.40	35	1.67
4	39.20	36	2.22
5	41.00	37	2.78
6	42.80	38	3.33
7	44.60	39	3.89
8	46.40	40	4.44
9	48.20	41	5.00
10	50.00	42	5.56

Bài giải: xem trang 91

Bài 30: Viết chương trình nhập lãi suất năm r (%), tiền vốn p và thời hạn gửi tiền n (năm). Mỗi trị nhập phải cách nhau bởi dấu “,”. In ra vốn tích lũy a của từng năm. Chương trình có kiểm tra nhập thiếu hoặc nhập lỗi.



$a = p(1 + r)^n$
 Trong đó, a (mount) là vốn tích lũy được, p (principal) là vốn gốc, r là (rate) lãi suất và n là số năm đầu tư.

⁴ Gabriel Fahrenheit (1686 - 1736) và Anders Celsius (1701 - 1744)



```
Nhap lai suat, tien von, thoi han: 0.027, 15000, 3 ↵
Lai suat: 2.7%
Von ban dau: 15000
Thoi han: 3 nam
Nam      Von
1       15405
2       15820.9
3       16248.1
```

Bài giải: xem trang 91

Bài 31: Viết chương trình in bảng cửu chương từ 2 đến 9 ra màn hình.



```
Bang cuu chuong
|2x 1= 2|3x 1= 3|4x 1= 4|5x 1= 5|6x 1= 6|7x 1= 7|8x 1= 8|9x 1= 9|
|2x 2= 4|3x 2= 6|4x 2= 8|5x 2=10|6x 2=12|7x 2=14|8x 2=16|9x 2=18|
|2x 3= 6|3x 3= 9|4x 3=12|5x 3=15|6x 3=18|7x 3=21|8x 3=24|9x 3=27|
|2x 4= 8|3x 4=12|4x 4=16|5x 4=20|6x 4=24|7x 4=28|8x 4=32|9x 4=36|
|2x 5=10|3x 5=15|4x 5=20|5x 5=25|6x 5=30|7x 5=35|8x 5=40|9x 5=45|
|2x 6=12|3x 6=18|4x 6=24|5x 6=30|6x 6=36|7x 6=42|8x 6=48|9x 6=54|
|2x 7=14|3x 7=21|4x 7=28|5x 7=35|6x 7=42|7x 7=49|8x 7=56|9x 7=63|
|2x 8=16|3x 8=24|4x 8=32|5x 8=40|6x 8=48|7x 8=56|8x 8=64|9x 8=72|
|2x 9=18|3x 9=27|4x 9=36|5x 9=45|6x 9=54|7x 9=63|8x 9=72|9x 9=81|
|2x10=20|3x10=30|4x10=40|5x10=50|6x10=60|7x10=70|8x10=80|9x10=90|
```

Bài giải: xem trang 92

Bài 32: Cho n_i là một số nguyên dương, với định nghĩa:

$$n_{i+1} = \begin{cases} n_i / 2 & n_i = 2k + 1 \\ 3n_i + 1 & n_i = 2k \end{cases}$$

Chuỗi trên sẽ ngừng khi n_i có trị 1. Các số được sinh ra gọi là hailstones (mưa đá) và quá trình trên đã được chứng minh là luôn luôn dừng. Viết chương trình sinh ra chuỗi hailstones với n_i nhập vào từ bàn phím.



```
Nhap n: 15 ↵
15 46 23 70 35 106
53 160 80 40 20 10
5 16 8 4 2 1
Hailstones sinh duoc: 18
Tiep (y/n)? n ↵
```

Bài giải: xem trang 93

Bài 33: Số tự nhiên có n chữ số là một số Armstrong (còn gọi là narcissistic numbers hoặc pluperfect digital invariants - PPDİ) nếu tổng các lũy thừa bậc n của các chữ số của nó bằng chính nó. Hãy tìm tất cả các số Armstrong có 3, 4 chữ số.

Ví dụ: 153 là số Armstrong có 3 chữ số vì: $1^3 + 5^3 + 3^3 = 153$



```
So Armstrong co 3, 4 chu so:
153 370 371 407 1634 8208 9474
```

Bài giải: xem trang 93

Bài 34: Dùng công thức hình thang, tính gần đúng tích phân xác định sau với độ chính xác 10^{-6} :

$$\int_0^{\pi/2} \sin^2(x) \cos(x) dx$$

Kiểm chứng với cách tính trực tiếp:

$$\int_0^{\pi/2} \sin^2(x) \cos(x) dx = \left. \frac{\sin^3(x)}{3} \right|_0^{\pi/2} = \frac{1}{3} \left(\sin^3\left(\frac{\pi}{2}\right) - \sin^3(0) \right) = \frac{1}{3} \sin^3\left(\frac{\pi}{2}\right)$$



Để tính gần đúng tích phân xác định, người ta thường dùng công thức hình thang (trapezoidal rule) như sau:

$$\begin{aligned} \int_a^b f(x) dx &\approx h \left[\frac{f(x_0) + f(x_1)}{2} + \frac{f(x_1) + f(x_2)}{2} + \dots + \frac{f(x_{n-1}) + f(x_n)}{2} \right] \\ &= h \left[\frac{f(x_0) + f(x_n)}{2} + f(x_1) + \dots + f(x_{n-1}) \right] = I_n \end{aligned}$$

$$\text{với: } h = \frac{b-a}{n}, x_i = a + ih$$



Để đạt độ chính xác, chọn n_0 tùy ý, sau đó tính I_n với $n = n_0, 2n_0, 4n_0 \dots$. Việc tính toán dừng lại khi $|I_{2n} - I_n|/3 < \epsilon$ (ϵ là độ chính xác).



Kết quả : 0.333333
Đôi chung: 0.333333

Bài giải: xem trang 94

Bài 35: Viết chương trình kiểm tra một số nguyên dương n có là số nguyên tố hay không. Nếu không thì phải xác định số nguyên tố gần n nhất và bé hơn n .



Số nguyên tố n là một số nguyên lớn hơn 1, chỉ có hai ước số (chỉ chia hết): 1 và chính nó.



Để xác định n là số nguyên tố, chỉ cần kiểm tra n không có ước số từ 2 đến \sqrt{n} ; do mọi hợp số (số nguyên lớn hơn 1 không phải là số nguyên tố) n đều có ước số nguyên tố nhỏ hơn \sqrt{n} ⁵.



Nhap n: 822 ↵
822 không là số nguyên tố
Số nguyên tố bé hơn gần nhất: 821

Bài giải: xem trang 95

Bài 36: Viết chương trình in ra n số nguyên tố đầu tiên (n nhập từ bàn phím).



Nhap n: 15 ↵
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

Bài giải: xem trang 97

⁵ Vì n là hợp số, ta có thể viết $n = a \cdot b$, trong đó a, b là các số nguyên với $1 < a \leq b < n$. Rõ ràng phải có a hoặc b không vượt quá \sqrt{n} , giả sử là b . Ước số nguyên tố của b cũng là ước số nguyên tố của n .

Bài 37: Viết chương trình nhập số nguyên dương n . Tìm số nguyên dương m lớn nhất sao cho: $1 + 2 + \dots + m < n$.



```
Nhap n: 22 ↵
1 + 2 + 3 + 4 + 5 + 6 = 21 < 22
m = 6
```

Bài giải: xem trang 98

Bài 38: Nhập vào một số tiền n (nghìn đồng, $n > 5$) nguyên dương. Đổi số tiền này ra ba loại tiền giấy 1000VNĐ, 2000VNĐ, 5000VNĐ.

Tìm phương án đổi tiền sao cho loại tiền 2000VNĐ chiếm hơn phân nửa số tờ bạc phải đổi ít nhất.



```
Nhap n (nghin dong, n > 5): 137 ↵
( 0, 21, 19 ): 40
```

Bài giải: xem trang 99

Bài 39: Tìm các bộ số nguyên dương (x, y, z) ($x, y, z < 100$) là 3 số nguyên liên tiếp hoặc 3 số chẵn liên tiếp, thỏa mãn công thức Pythagoras⁶: $x^2 + y^2 = z^2$



```
(3, 4, 5): ba so nguyen lien tiep
(6, 8, 10): ba so chan lien tiep
```

Bài giải: xem trang 99

Bài 40: Tìm các bộ (trâu đứng, trâu nằm, trâu già) thỏa mãn bài toán cổ:

Trăm trâu ăn trăm bó cỏ

Trâu đứng ăn năm

Trâu nằm ăn ba

Lục khụ trâu già

Ba con một bó

Thử tìm cách giảm số vòng lặp khi tính toán xuống.



```
(4, 18, 78)
(8, 11, 81)
(12, 4, 84)
```

Bài giải: xem trang 100

Bài 41: Viết chương trình tìm cách thay thế các dấu hỏi (?) bởi các dấu 4 phép tính số học +, -, *, /, trong biểu thức dưới đây sao cho biểu thức có giá trị bằng 36.

$(((((1 ? 2) ? 3) ? 4) ? 5) ? 6$



```
(((((1 - 2) + 3) + 4) * 5) + 6 = 36
(((1 - 2) * 3) + 4) + 5) * 6 = 36
(((1 * 2) + 3) - 4) + 5) * 6 = 36
(((1 / 2) * 3) * 4) * 5) + 6 = 36
```

Bài giải: xem trang 100

⁶ Pythagoras (582 BC - 507 BC)

Bài 42: Giả thuyết Goldbach⁷ (Goldbach's conjecture) cho rằng: một số nguyên tố n bất kỳ ($n > 5$) đều có thể khai triển thành tổng của ba số nguyên tố khác.

Viết chương trình kiểm chứng giả thuyết Goldbach với $n < 1000$.



```
Co 165 so nguyen to n (5 < n < 1000)
  7 = 2 + 2 + 3
 11 = 2 + 2 + 7
    ...
 997 = 3 + 3 + 991
Kiểm chứng đúng với 165 số nguyên tố
```

Bài giải: xem trang 101

Bài 43: Tìm số Fibonacci⁸ thứ n ($n < 40$), dùng vòng lặp (không dùng đệ quy).



Số Fibonacci thứ n : $F(n) = \begin{cases} 1 & n = 1, 2 \\ F(n-2) + F(n-1) & n > 2 \end{cases}$



```
Nhap n (n < 40): 24 ↵
Fi(24) = 46368
```

Bài giải: xem trang 102

Bài 44: Dùng vòng lặp lồng, viết chương trình in ra tam giác cân đặc và rỗng, tạo từ các dấu sao (*), có độ cao là n nhập từ bàn phím.



```
Nhap n: 4 ↵
  *
 * * *
* * * * *
 *
  *
*      *
 * * * * *
  *
 * * * * *
```

Bài giải: xem trang 103

Bài 45: Dùng vòng lặp lồng, với n ($n < 5$) nhập từ bàn phím, viết chương trình in hai tam giác đối đỉnh bằng số, tăng theo cột từ 1 đến $2n - 1$.



```
Nhap n (n < 5): 3 ↵
  1      5
 1 2    4 5
1 2 3 4 5
 1 2    4 5
  1      5
```

Bài giải: xem trang 104

Bài 46: Viết chương trình kiểm tra hai vế của công thức sau, với n cho trước:

$$\sum_{i=3}^n i^3 = \frac{n^2(n+1)^2}{4}$$

⁷ Christian Goldbach (1690 - 1764)

⁸ Leonardo Fibonacci (1170 - 1250)



```
Nhap n: 50 ↵
Ve trai = 1625625
Ve phai = 1625625
```

Bài giải: xem trang 104

Bài 47: Với n cho trước, tính tổng S , biết:

Nếu n chẵn: $S = 2 + 4 + 6 + \dots + n$

Nếu n lẻ: $S = 1 + 2 + 3 + \dots + n$



```
Nhap n: 120 ↵
S = 3660
```

Bài giải: xem trang 105

Bài 48: Với số nguyên n cho trước, tìm ước số lẻ lớn nhất của n và ước số lớn nhất của n là lũy thừa của 2.



```
Nhap n: 384 ↵
US le lon nhat: 3
US lon nhat la luy thua cua 2: 128
```

Bài giải: xem trang 105

Bài 49: Viết chương trình tính căn số liên tục sau:

$$S = \sqrt[n+1]{n + \sqrt[n]{n-1 + \sqrt[n-1]{n-2 + \dots + \sqrt[3]{2 + \sqrt{1}}}}}$$



```
Nhap n: 10 ↵
Ket qua: 1.24624
```

Bài giải: xem trang 107

Bài 50: Phân số liên tục (continued fraction) ký hiệu $[b_1, b_2, \dots, b_k]$, có dạng:

$$\frac{s}{t} = \frac{1}{b_1 + \frac{1}{b_2 + \frac{1}{\dots + \frac{1}{b_{k-1} + \frac{1}{b_k}}}}}$$

b_1, b_2, \dots, b_k là các số tự nhiên. Cho s và t , viết chương trình tìm $[b_1, b_2, \dots, b_k]$.



Mỗi phân số hữu tỷ $\frac{s}{t}$ ($0 < s < t$ là các số tự nhiên) đều có thể đưa về dạng phân số liên tục bằng thuật toán sau:

1. Chia t cho s , được a dư r : $t = a * s + r$. Suy ra: $\frac{s}{t} = \frac{1}{\frac{t}{s}} = \frac{1}{a + \frac{r}{s}}$
2. Đặt $b_1 = a$, rồi tiếp tục biến đổi $\frac{r}{s}$ cho đến khi số dư r bằng 0.



```
Nhap s, t (0 < s < t): 123 1234 ↵
[10, 30, 1, 3]
```

Bài giải: xem trang 107

Bài 51: Viết chương trình tính phân số liên tục sau:

$$F = x + \frac{1}{x + \frac{2}{x + \frac{4}{\dots x + \frac{128}{x + \frac{256}{x}}}}}$$

(x là số thực khác 0)



Nhap x: 2.4 ↵
F = 2.73649

Bài giải: xem trang 108

Bài 52: Cho số tự nhiên n, hãy tính F_n biết: $F_n = \sum_{i=1}^n \frac{1}{n^2 + i}$



Nhap n: 12 ↵
Fn = 0.0797762

Bài giải: xem trang 109

Bài 53: Viết chương trình tính $\sin(x)$ với độ chính xác 10^{-4} theo chuỗi Taylor⁹ (Taylor series):

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$



Nhap x (radian): 2.7 ↵
cong thuc Taylor: $\sin(2.70) = 0.4274$
 $\sin()$ của math.h: $\sin(2.70) = 0.4274$

Bài giải: xem trang 109

Bài 54: Dùng vòng lặp, tính tổ hợp chập k của n ($k < n < 25$):

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Kiểm chứng công thức $C_n^k = C_n^{n-k}$



Nhap n, k ($k < n < 25$): 20 5 ↵
C(k, n): 15504
C(n-k, n): 15504

Bài giải: xem trang 110

Bài 55: Tính căn bậc hai của một số nguyên dương x bằng thuật toán Babylonian. Kiểm tra kết quả với hàm chuẩn $\text{sqrt}()$.



Thuật toán Babylonian dùng tính căn bậc hai của một số nguyên dương x:

1. Đặt $y = 1.0$
2. Thay y với trung bình cộng của y và x/y
3. Lặp lại bước 2 đến khi y không còn thay đổi (y xấp xỉ bằng x/y)
4. Trả về y

⁹ Brook Taylor (1685 - 1731)



Nhap x ($x > 0$): 7 ↵
 thuật toán babylonian: 2.64575
 hàm sqrt() của math.h: 2.64575

Bài giải: xem trang 111

Bài 56: Viết chương trình nhập vào một số nguyên n có dấu, in ra dạng hiển thị nhị phân và thập lục phân của n .



Để xác định một bit tại vị trí bất kỳ, dùng mặt nạ (mask) AND, kết hợp với toán tử AND bitwise (&):

Mặt nạ thường là một dãy bit 0, với bit 1 được bật tại vị trí cần kiểm tra.

	10	1	1010
mask	00	1	0000
	00	1	0000

$(\neq 0) \Rightarrow \text{bit } 1$

	10	0	1010
mask	00	1	0000
	00	0	0000

$(=0) \Rightarrow \text{bit } 0$

Khi cần xác định bit tại vị trí khác, dùng toán tử dịch bit để di chuyển bit 1 của mặt nạ; hoặc dịch chuyển số kiểm tra để bit cần kiểm tra đến đúng vị trí bit 1 của mặt nạ.

Không dùng cách chia 2 để chuyển số nguyên thành số nhị phân rồi lấy từng bit vì số nguyên âm lưu ở dạng số bù 2 (two's complement).



Nhap n : -5678 ↵
 -5678 = 11111111 11111111 11101001 11010010
 Hex: FFFFE9D2

Bài giải: xem trang 111

Bài 57: Bit parity là bit thêm vào một nhóm mã được truyền đi, dùng để phát hiện lỗi một bit đơn trong quá trình truyền. Bit parity chẵn (even parity) là bit có trị được chọn sao cho tổng số bit 1 trong một nhóm mã kể cả bit parity là một số chẵn. Viết chương trình nhập vào một số nguyên n . Xác định bit parity chẵn của n .



Bit parity chẵn của n sẽ bằng 0 nếu số các bit 1 là số chẵn và bằng 1 nếu số các bit 1 là số lẻ.



Nhap n : 13579 ↵
 Even parity bit = 1



Bài giải: xem trang 112

MẢNG

Bài 58: Viết chương trình thực hiện thuật toán sàng Eratosthenes¹⁰ (Sieve of Eratosthenes) để in ra các số nguyên tố nhỏ hơn số n cho trước ($n < 100$).



Sàng Eratosthenes: viết các số nguyên từ 2 đến n . Khoanh tròn 2; gạch chéo tất cả những bội số khác của 2. Lặp lại bằng cách khoanh tròn số nhỏ nhất chưa được khoanh tròn, không bị gạch chéo; gạch chéo tất cả những

¹⁰ Eratosthenes (276 BC - 194 BC)

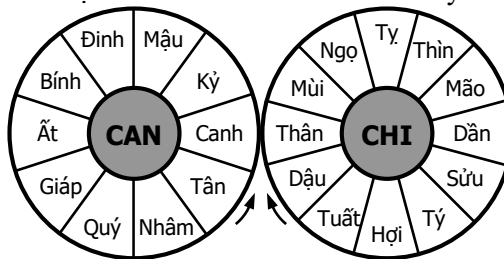
bội số của nó. Khi không còn số nào để khoanh tròn hoặc gạch chéo thì dừng. Tất cả những số được khoanh tròn là số nguyên tố.



Nhap n: 64 ↵
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61

Bài giải: xem trang 113

Bài 59: Nhập vào năm Dương lịch, xuất tên năm Âm lịch. Xuất năm Dương lịch kế tiếp có cùng tên năm Âm lịch. Biết bánh xe tính hai chu kỳ Can - Chi như sau:



Năm có cùng tên Âm lịch với năm y là $y \pm k * 60$ (60 là BSCNN của hai chu kỳ 10 và 12). Mốc tính Can Chi, lấy năm 0 là năm Canh Thân.



Nhap nam: 2000 ↵
2000 - Canh Thìn
2060 - Canh Thìn

Bài giải: xem trang 114

Bài 60: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên (n chẵn) có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Viết hàm thực hiện việc trộn hoàn hảo (perfect shuffle) một mảng: sao cho các phần tử của một nửa mảng sau xen kẽ với các phần tử của một nửa mảng đầu. Xuất mảng sau khi trộn.
- Xác định số lần trộn hoàn hảo để mảng trở về như ban đầu.



Nhap n (n chan): 12 ↵
-33 62 -12 34 -89 65 -3 -96 86 89 39 35
-33 -3 62 -96 -12 86 34 89 -89 39 65 35
Can 10 lan shuffle de mang tro ve ban dau

Bài giải: xem trang 114

Bài 61: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Tính tổng các số nguyên dương có trong mảng.
- Xóa phần tử có chỉ số p (p nhập từ bàn phím) trong mảng.



Nhap n [1, 99]: 10 ↵
69 -41 48 22 -34 100 -14 70 66 -29
Tong cac so nguyen duong = 375
Nhap p [0, 9]: 4 ↵

69 -41 48 22 100 -14 70 66 -29

Bài giải: xem trang 116**Bài 62:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên dương có giá trị chứa trong đoạn $[10, 20]$ và xuất mảng.
- Kiểm tra xem tổng các số chẵn ở vị trí lẻ có bằng tổng các số lẻ ở vị trí chẵn hay không?
- Xác định xem mảng có cặp số nguyên tố cùng nhau (coprime) nào không.



Hai số nguyên dương a và b được gọi là hai số nguyên tố cùng nhau nếu ước số chung lớn nhất của hai số a và b là 1.



```
Nhap n [1, 99]: 5 ↵
14 14 11 16 12
Tong le vi tri chan (30) khac tong chan vi tri le (11)
Cac cap nguyen to cung nhau:
(14, 11)
(11, 16)
(11, 12)
```

Bài giải: xem trang 117**Bài 63:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Đếm số phần tử chia hết cho 4 và có chữ số tận cùng là 6.
- Thay các phần tử lẻ bằng 2 lần giá trị của nó.



```
Nhap n [1, 99]: 10 ↵
70 -67 22 -87 34 16 -34 -58 76 -78
Co 2 phan tu chia het cho 4, tan cung 6
Nhan doi phan tu le:
70 -134 22 -174 34 16 -34 -58 76 -78
```

Bài giải: xem trang 118**Bài 64:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Hãy đếm số các phần tử có trị là lũy thừa của 2 có trong mảng.
- Nhập x nguyên, xóa các phần tử trong mảng có trị trùng với x .



Một số là lũy thừa của 2 nếu số đó có bit 1 duy nhất là bit MSB (Most Significant Bit). Ví dụ: $2^8 = 256 = 1\ 0000\ 0000$



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
2 -5 4 7 9 -8 32 16 11 4 ↵
Co 5 so la luy thua cua 2
Nhap x: 4 ↵
2 -5 7 9 -8 32 16 11
```

Bài giải: xem trang 119

Bài 65: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Tính trung bình cộng của các số nguyên âm lẻ có trong mảng.
- Xóa các phần tử có trị trùng nhau trong mảng, chỉ chừa lại một phần tử.



```
Nhap n [1, 99]: 10 ↵
Nhap 10phan tu:
2 2 -3 7 4 -5 4 9 -1 -1 ↵
Trung binh cong nguyen am le = -2.50
2 -3 7 4 -5 9 -1
```

Bài giải: xem trang 120

Bài 66: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Dùng một vòng lặp, tìm phần tử có trị nhỏ nhất và lớn nhất của mảng.
- Xóa các phần tử trong mảng có trị trùng với giá trị lớn nhất của mảng, trừ phần tử tìm được đầu tiên.



```
Nhap n [1, 99]: 10 ↵
21 1 -68 24 22 -76 -69 0 24 -84
max = 24
min = -84
21 1 -68 24 22 -76 -69 0 -84
```

Bài giải: xem trang 121

Bài 67: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Sắp xếp sao cho các vị trí chứa trị chẵn trên mảng vẫn chứa trị chẵn nhưng có thứ tự tăng, các vị trí chứa trị lẻ trên mảng vẫn chứa trị lẻ nhưng có thứ tự giảm.



```
Nhap n [1, 99]: 10 ↵
72 -8 45 -97 77 25 -86 86 -2 60
-86 -8 77 45 25 -97 -2 60 72 86
```

Bài giải: xem trang 122

Bài 68: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên (n chẵn) có giá trị chứa trong đoạn $[100, 200]$ và xuất mảng.
- Chia các phần tử của mảng thành hai nhóm, sao cho hiệu của tổng các phần tử nhóm này và tổng các phần tử nhóm kia là một số dương nhỏ nhất.



Tìm cặp a_0, b_0 ($a_0 > b_0$) có hiệu nhỏ nhất, cặp a_1, b_1 ($a_1 > b_1$) có hiệu nhỏ thứ hai, ... Như vậy hiệu $(a_0 + a_1 + \dots) - (b_0 + b_1 + \dots)$ sẽ nhỏ nhất.
Tham khảo thêm bài 62, trang 19.



```
Nhap n (n chan): 10 ↵
109 111 162 107 115 111 108 173 108 113
```

```
111 108 109 115 173 : 616
111 108 107 113 162 : 601
Hieu nho nhat = 15
```

Bài giải: xem trang 124

Bài 69: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Xuất ra màn hình “run” tăng dài nhất tìm thấy đầu tiên.



“run” là chuỗi các phần tử (liên tục) theo cùng một quy luật nào đó (tăng dần, giảm dần, chẵn, lẻ, bằng nhau, ...).



```
Nhap n [1, 99]: 10 ↵
-53 -32 23 78 61 -1 95 83 -55 -7
"run" tang dai nhat: -53 -32 23 78
```

Bài giải: xem trang 126

Bài 70: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Hãy chuyển các phần tử có trị lẻ về đầu mảng, các phần tử có trị chẵn về cuối mảng. Các phần tử có trị 0 nằm ở giữa.



```
Nhap n [1, 99]: 10
-66 64 0 -50 58 51 0 45 1 82
51 45 1 0 0 -50 58 -66 64 82
```

Bài giải: xem trang 127

Bài 71: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Kiểm tra xem mảng có đối xứng hay không.
- Hãy dịch trái xoay vòng mảng k lần, k nhập từ bàn phím.



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
1 2 3 4 5 5 4 3 2 1 ↵
Doi xung
Nhap so lan can dich: 3 ↵
4 5 5 4 3 2 1 1 2 3
```

Bài giải: xem trang 128

Bài 72: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Kiểm tra trong mảng có số lẻ hay không? Nếu có tìm số lẻ lớn nhất.
- Hãy dịch phải xoay vòng mảng k lần, k nhập từ bàn phím.



```
Nhap n [1, 99]: 10 ↵
4 -33 36 -4 12 72 -9 -87 76 -40
Phan tu le lon nhat: a[6] = -9
```

```
Nhap so lan can dich: 3 ↵
-87 76 -40 4 -33 36 -4 12 72 -9
```

Bài giải: xem trang 129

Bài 73: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- In ra các phần tử trong mảng có trị phân biệt.



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
1 2 2 3 4 3 1 5 5 4 ↵
1 2 3 4 5
```

Bài giải: xem trang 131

Bài 74: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Thống kê số lần xuất hiện của các phần tử trong mảng.



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
1 2 2 3 4 3 2 5 5 3 ↵
1[1] 2[3] 3[3] 4[1] 5[2]
```

Bài giải: xem trang 132

Bài 75: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Xuất phần tử xuất hiện nhiều nhất, xuất hiện ít nhất tìm thấy đầu tiên.



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
3 2 2 3 4 3 2 5 5 3 ↵
Phan tu xuat hien nhieu nhat: 3[4]
Phan tu xuat hien it nhat: 4[1]
```

Bài giải: xem trang 133

Bài 76: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Tìm các phần tử có số lần xuất hiện là như nhau và nhiều nhất.



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu:
1 2 2 3 4 4 2 5 5 4 ↵
Phan tu xuat hien nhieu nhat:
2[3] 4[3]
```

Bài giải: xem trang 134

Bài 77: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Tính tổng nghịch đảo các phần tử trong mảng.

- c. Viết hàm duyệt các phần tử $A[i]$ của mảng theo thứ tự từ trái sang phải; nếu $A[i]$ lẻ thì xóa một phần tử bên phải nó.



```
Nhap n [1, 99]: 10 ↵
-1 -39 62 -48 -12 -32 -39 87 75 -53
Tong nghich dao: -1
-1 62 -48 -12 -32 -39 75
```

Bài giải: xem trang 135

Bài 78: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Hãy sắp xếp các phần tử trong mảng theo thứ tự tăng dần.
- Hãy chèn một phần tử x vào trong mảng đã được sắp tăng dần mà vẫn giữ nguyên tính tăng dần của nó.



```
Nhap n [1, 99]: 10 ↵
-94 63 -78 2 7 -26 -82 8 -18 39
Mang sap xep tang:
-94 -82 -78 -26 -18 2 7 8 39 63
Nhap x: 0 ↵
-94 -82 -78 -26 -18 0 2 7 8 39 63
```

Bài giải: xem trang 136

Bài 79: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng.
- Nhập số nguyên x , tìm phần tử trong mảng gần x nhất.
- Viết hàm chèn 1 vào bên phải các phần tử có trị âm của mảng.



```
Nhap n (n > 0): 10 ↵
55 98 87 93 -37 -50 77 -48 93 52
Nhap x: 50
So gan x nhat: 52 ↵
55 98 87 93 -37 1 -50 1 77 -48 1 93 52
```

Bài giải: xem trang 138

Bài 80: Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều A , n phần tử nguyên có giá trị nhập vào từ bàn phím.
- Tạo mảng một chiều B , m phần tử nguyên ($m \leq n$), có giá trị nhập vào từ bàn phím. Tìm vị trí xuất hiện đầu tiên của mảng B trong mảng A .
- Tìm số nguyên âm cuối cùng của mảng A .



```
Nhap n [1, 99]: 10 ↵
Nhap 10 phan tu mang A:
-1 2 -3 4 -5 5 -4 3 -2 1 ↵
Nhap m [1, 10]: 4 ↵
Nhap 4 phan tu mang B:
4 -5 5 -4 ↵
B co trong A tai: A[3]
So nguyen am cuoi: -2
```

Bài giải: xem trang 140

Bài 81: Viết hàm trộn hai mảng A, B đã được sắp xếp tăng, sao cho mảng kết quả là một mảng sắp xếp giảm. Không được sắp xếp trực tiếp mảng kết quả.



```
Nhap so phan tu mang A va B (n, m > 0): 5 7 ↵
17 -21 0 100 -42
67 11 -66 32 52 22 -48
Mang A sap tang: -42 -21 0 17 100
Mang B sap tang: -66 -48 11 22 32 52 67
Tron A va B thanh C sap giam:
100 67 52 32 22 17 11 0 -21 -42 -48 -66
```

Bài giải: xem trang 141

Bài 82: Viết chương trình cho phép người dùng nhập n số tùy ý, nhập cho đến khi nhấn Ctrl+Z. Hãy lưu các số này thành một tập hợp chứa các phần tử có trị phân biệt.

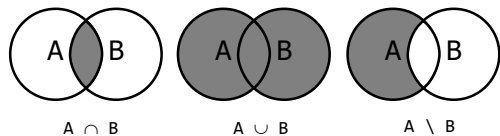


```
Nhap khong qua 100 phan tu (Ctrl+Z de dung)
1 3 5 7 2 4 3 6 7 5 4 8 ↵
^Z ↵
Tap hop A: {1, 3, 5, 7, 2, 4, 6, 8}
```

Bài giải: xem trang 143

Bài 83: Cho hai mảng A, B là hai tập hợp, khởi tạo trước hoặc nhập từ bàn phím. Tạo mảng C là một tập hợp gồm các phần tử:

- Xuất hiện trong cả A và B (giao).
- Không xuất hiện trong B (hiệu).
- Xuất hiện trong A hoặc B (hợp).



```
Tap hop A: {1, 2, 3, 5}
Tap hop B: {1, 3, 6, 7}
C = A * B: {1, 3}
C = A + B: {1, 2, 3, 5, 6, 7}
C = A \ B: {2, 5}
```

Bài giải: xem trang 144

Bài 84: Viết chương trình nhập thực hiện những yêu cầu sau:

- Viết hàm chèn từng phần tử vào một mảng số nguyên sao cho mảng luôn giữ thứ tự giảm.
- Dùng hàm này để lưu các trị nhập thành một mảng có thứ tự giảm. Nhập cho đến khi nhấn Ctrl+Z, xuất mảng để kiểm tra.



```
Nhap khong qua 100 phan tu (Ctrl+Z de dung):
3 5 4 7 2 6 9 1 8 ↵
^Z ↵
9 8 7 6 5 4 3 2 1
```

Bài giải: xem trang 145

MẢNG CỦA CÁC MẢNG¹¹**Bài 85:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng hai chiều vuông có kích thước 4×4 với các phần tử ngẫu nhiên trong đoạn $[-100, 100]$ và xuất mảng.
- Sắp xếp lại các phần tử của mảng hai chiều trên sao cho mỗi dòng tăng từ trái sang phải và mỗi cột tăng từ trên xuống dưới.



Mang goc:

100	21	-67	-81
1	55	31	-61
48	99	62	3
51	27	-61	14

Mang sau khi sap xep:

-81	-67	-61	-61
1	3	14	21
27	31	48	51
55	62	99	100

Bài giải: xem trang 146**Bài 86:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng hai chiều vuông có kích thước $n \times n$ (n nhập từ bàn phím) với các phần tử ngẫu nhiên trong đoạn $[-100, 100]$ và xuất mảng.
- Sắp xếp lại các cột của mảng hai chiều trên sao cho tổng trị các phần tử của mỗi cột tăng dần từ trái sang phải.

Nhap n ($n < 20$): 3 ↵

-48	61	31
-22	-23	-62
83	-24	-18

Mang sau khi sap xep:

61	31	-48
-23	-62	-22
-24	-18	83

Bài giải: xem trang 148**Bài 87:** Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A vuông bậc n (n nhập từ bàn phím) với các phần tử được nhập từ bàn phím, xuất ma trận.
- Tính tổng các phần tử trên đường chéo chính (vết - trace) của ma trận A.
- Kiểm tra xem ma trận A có phải là ma trận tam giác trên hay không. Nếu phải, tính định thức của ma trận đó.



Ma trận tam giác trên là ma trận có các phần tử nằm dưới đường chéo chính đều bằng 0. Định thức của ma trận tam giác trên theo Gauss¹²:

$$\det(A) = \prod_{i=1}^n A_{ii}$$

¹¹ C không hỗ trợ mảng nhiều chiều (ví dụ $a[5,7]$), C chỉ hỗ trợ mảng có kiểu bất kỳ, kể cả mảng chứa các mảng (ví dụ $a[5][7]$). Để dễ trình bày, chúng tôi vẫn dùng thuật ngữ mảng 1 chiều, 2 chiều, ...

¹² Carl Friedrich Gauss (1777 - 1855)



```
Nhap bac ma tran: 3 ↵
a[0][0] = 1 ↵
...
a[2][2] = 9 ↵
  1   2   3
  0   5   6
  0   0   9
Trace = 15
det(A) = 45
```

Bài giải: xem trang 149

Bài 88: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Viết hàm kiểm tra xem ma trận có đồng nhất hay không. Nếu không, in ra ma trận đồng nhất cùng bậc với ma trận trên.



Ma trận đồng nhất I_n (identity matrix) là ma trận vuông bậc n có các phần tử đều bằng 0, trừ các phần tử trên đường chéo chính đều bằng 1.



```
Nhap bac ma tran: 3 ↵
  43  65  29
  78 -67  26
 -72 -61  95
Ma tran tren khong dong nhat
  1   0   0
  0   1   0
  0   0   1
```

Bài giải: xem trang 150

Bài 89: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Hoán chuyển phần tử lớn nhất nằm trên mỗi dòng với phần tử nằm trên đường chéo chính cũng thuộc dòng đó.



```
Nhap bac ma tran: 3 ↵
 -27  11 -81
 -10 -13  35
 -24  61 -17
Ma tran sau khi sap xep:
  11 -27 -81
 -10  35 -13
 -24 -17  61
```

Bài giải: xem trang 151

Bài 90: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.

- b. Viết hàm xuất tất cả các đường chéo song song với đường chéo phụ, mỗi đường chéo trên một dòng.



```
Nhap bac ma tran: 3 ↵
-84  50  68
 53 -94 -47
-62 -59  67

Cac duong cheo song song cheo phu:
-84
 50  53
 68 -94 -62
-47 -59
 67
```

Bài giải: xem trang 152

Bài 91: Một mẫu lát (pattern) thường được định nghĩa dưới dạng một ma trận nhị phân (xem hình dưới). Giả sử mẫu lát trên dùng lát kín ma trận A (kích thước 20×20), xuất ma trận B là ma trận con của A, kích thước 8×12 , có góc trên trái nằm tại phần tử A[3][4].

*	*
*	*
*	*
*	*

0	1	0	1	0	0
1	0	0	0	1	0
1	0	0	0	1	0
0	1	0	1	0	0



```

* * * *
* * * *
* * * *
* * * *
  * * * *
  * * * *
* * * *
* * * *
```

Bài giải: xem trang 153

Bài 92: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Tìm đường chéo, song song với đường chéo chính và có tổng trị các phần tử là lớn nhất.



```
Nhap bac ma tran: 3 ↵
-99 -24 -89
 97 -66  10
 16  22 -66

Duong cheo co tong lon nhat:
97 22 : 119
```

Bài giải: xem trang 154

Bài 93: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A vuông bậc 3 với các phần tử là các trị ngẫu nhiên trong đoạn $[0, 9]$, xuất ma trận.
- Với k nguyên dương nhập từ bàn phím ($k > 1$), xuất ma trận lũy thừa A^k .



Nếu A là một ma trận $n \times n$ và nếu k là một số nguyên dương thì A^k là ký hiệu của tích k lần ma trận A, gọi là ma trận lũy thừa A^k . $A^0 = I$ (ma trận đồng nhất). Ma trận lũy thừa thường dùng khi xác định tính liên thông của đồ thị bằng ma trận kề. Các phần tử của ma trận tích $C = A.B$ là $C_{ij} = \sum_{k=1}^n A_{ik}B_{kj}$. Tích A.B tồn tại khi số cột của A bằng số dòng của B.



```

3      8      8
9      2      8
9      3      6

Nhập lũy thừa (k > 1): 3 ↵
2259 1760 2552
2475 1544 2552
2376 1452 2424

```

Bài giải: xem trang 155

Bài 94: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A vuông bậc n ($n > 1$, nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Tạo rồi xuất ma trận con B vuông bậc $n - 1$ từ ma trận A bằng cách loại bỏ dòng và cột chứa phần tử có trị tuyệt đối lớn nhất.



```

Nhập bậc ma trận (n > 1): 3 ↵
-56   42    5
-18   82   18
 50  -39   42

Ma trận B:
-56    5
 50   42

```

Bài giải: xem trang 157

Bài 95: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Cho k nguyên với $0 \leq k < n$. Tạo và xuất ma trận B vuông bậc n từ ma trận A bằng cách hoán chuyển dòng k với cột k.



```

Nhập bậc ma trận: 3 ↵
-52   31  -75
-92   59   27
-58   47   29

Nhập k: 1 ↵
-52  -92  -75
 31   59   47
-58   27   29

```

Bài giải: xem trang 159

Bài 96: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A vuông bậc n (n nhập từ bàn phím) với các phần tử có trị: $A_{ij} = \sin(i - 2 * j / \pi)$, xuất ma trận.

- b. Đếm số phần tử không âm của ma trận.



```
Nhap bac ma tran: 4 ↵
0.000000 -0.594481 -0.956056 -0.943067
0.841471 0.355436 -0.269852 -0.789417
0.909297 0.978566 0.664452 0.090019
0.141120 0.702007 0.987862 0.886692
Co 11 phan tu khong am
```

Bài giải: xem trang 159

Bài 97: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận A có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Lân cận của phần tử A_{ij} được hiểu là các phần tử có chỉ số tương ứng chênh lệch với i, j không quá 1 đơn vị. Xuất ma trận nhị phân B (gọi là ma trận cực tiểu) có kích thước $m \times n$ với $B_{ij} = 1$ khi tất cả những phần tử lân cận A_{ij} đều lớn hơn A_{ij} (khi đó A_{ij} được gọi là phần tử cực tiểu), các phần tử B_{ij} còn lại bằng 0.



```
Nhap n, m: 3 4 ↵
14 78 54 -43
84 71 -45 22
-54 35 14 83
Ma tran cuc tieu:
1 0 0 0
0 0 1 0
1 0 0 0
```

Bài giải: xem trang 160

Bài 98: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- In ma trận xoay 90° theo chiều kim đồng hồ từ của ma trận trên. Tâm quay là phần tử góc dưới trái.



```
Nhap n, m: 3 4 ↵
63 -49 88 95
-52 -94 -33 48
73 -69 86 -38
Ma tran sau khi quay:
73 -52 63
-69 -94 -49
86 -33 88
-38 48 95
```

Bài giải: xem trang 161

Bài 99: Viết chương trình thực hiện những yêu cầu sau:

- Tạo hai ma trận vuông cùng bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất hai ma trận.
- In ma trận tổng và ma trận tích của hai ma trận trên.



```
Nhap bac ma tran: 2 ↵
Ma tran A:
```

```

-10  9
  7  9
Ma tran B:
  3 -1
  7  4
Ma tran tong:
 -7  8
 14 13
Ma tran tich:
 33 46
 84 29

```

Bài giải: xem trang 163

Bài 100: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử được nhập từ bàn phím, xuất ma trận.
- Ma trận gọi là “thưa” (sparse matrix) nếu số phần tử có trị 0 nhiều hơn số phần tử có trị khác 0. Kiểm tra xem ma trận trên có “thưa” hay không.



```

Nhap n, m: 3 4 ↵
a[0][0] = 3 ↵
...
a[2][3] = 0 ↵
  3  0  2 -5
  0 -4  0  0
  0  0  0  0
Ma tran thua

```

Bài giải: xem trang 165

Bài 101: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử nhập vào từ bàn phím, xuất ma trận.
- Xuất tổng các hàng và tổng các cột.
- Kiểm tra xem ma trận trên có tạo thành một ma phương không.



Ma phương (magic square) là hình vuông có tổng đường chéo chính, đường chéo phụ, hàng bất kỳ, cột bất kỳ đều bằng nhau. Ví dụ, hình vuông bên phải là ma phương bậc 3 với tổng 15.

2	9	4
7	5	3
6	1	8



```

Nhap bac ma tran: 3 ↵
A[0][0] = 2 ↵
...
A[2][2] = 8 ↵
  2  9  4  Tong hang 0: 15  Tong cot 0: 15
  7  5  3  Tong hang 1: 15  Tong cot 1: 15
  6  1  8  Tong hang 2: 15  Tong cot 2: 15
Ma phuong

```

Bài giải: xem trang 166

Bài 102: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$, xuất ma trận.
- Chuyển ma trận A thành ma trận chuyển vị A^* và in ra.



Ma trận A^T kích thước $m \times n$ với $A^T_{ji} = A_{ij}$ được gọi là ma trận chuyển vị (transpose) của ma trận A kích thước $n \times m$.



```
Nhap n, m: 3 4 ↵
-31  79 -30  43
 60 -69  87 -36
 72  10 -63  53
Ma tran chuyen vi:
-31  60  72
 79 -69  10
-30  87 -63
 43 -36  53
```

Bài giải: xem trang 167

Bài 103: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Xuất ma trận B từ ma trận nguồn A , sao cho $B[i][j]$ bằng tổng các phần tử thuộc hai đường chéo đi qua phần tử $A[i][j]$.



```
Nhap bac ma tran: 3 ↵
-4  -6  -7
 7   8  -9
 4   7 -10
Ma tran B:
-6  -8   5
 8  -9  -8
 5   5  -6
```

Bài giải: xem trang 168

Bài 104: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Xuất ma trận B từ ma trận nguồn A , sao cho $B[i][j]$ bằng số lớn nhất trong hai đường chéo đi qua phần tử $A[i][j]$.



```
Nhap bac ma tran: 3 ↵
-3   3  -1
 1  -7  -4
 2  -5  -7
Ma tran B:
-3   3   2
 3   2   3
 2   1  -3
```

Bài giải: xem trang 170

Bài 105: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Tạo ma trận B từ ma trận nguồn A sao cho $B[i][j]$ bằng tổng các phần tử không thuộc dòng i cột j của ma trận A.

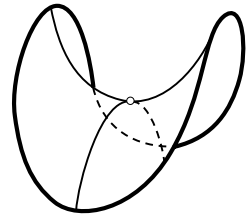


```
Nhap bac ma tran: 3 ↵
  9   3   0
-10  -8  -4
  -1   0   1
Ma tran B:
-11 -14 -19
  4   9  11
  -9  -5  -6
```

Bài giải: xem trang 170

Bài 106: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Tìm các điểm yên ngựa của ma trận trên.



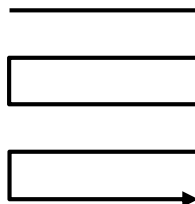
Phần tử A_{ij} gọi là *điểm yên ngựa* (saddle point) của ma trận A nếu nó vừa là phần tử nhỏ nhất của dòng i , đồng thời là phần tử lớn nhất của cột j ; hoặc ngược lại, vừa là phần tử lớn nhất của dòng i , đồng thời là phần tử nhỏ nhất của cột j .
Một ma trận có thể *không có* hoặc có nhiều điểm yên ngựa.



```
Nhap bac ma tran: 3 ↵
-2   5  -1
-3   1  -2
-6   2  -4
MIN dong MAX cot: a[0][0] = -2
MAX dong MIN cot: a[1][1] = 1
```

Bài giải: xem trang 171

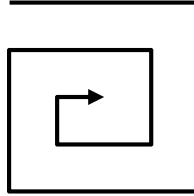
Bài 107: Tạo ma trận vuông bậc 5 với các phần tử có trị từ 1, 2, ..., 25, sắp xếp giảm theo hình zigzag từ trái sang phải, từ trên xuống dưới.



```
 1  2  3  4  5
10  9  8  7  6
11 12 13 14 15
20 19 18 17 16
21 22 23 24 25
```

Bài giải: xem trang 172

Bài 108: Tạo ma trận vuông bậc n (n nhập từ bàn phím, $n < 20$) với các phần tử có trị từ 1 đến n^2 sắp xếp tăng theo hình xoắn ốc từ trái sang phải, từ ngoài vào trong.



Nhap bac ma tran ($n < 20$): 5 ↵

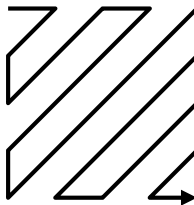
```

1  2  3  4  5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9

```

Bài giải: xem trang 173

Bài 109: Tạo ma trận vuông bậc 5 với các phần tử có trị từ 1, 2, ..., 25, xuất ma trận. Viết chương trình sắp xếp trị của các phần tử trong ma trận tăng theo thứ tự như hình dưới:



Thực chất là ghi lần lượt các phần tử lên các đường chéo song song với đường chéo phụ. Mỗi lần chuyển sang đường chéo mới thì lại đổi chiều ghi, thực chất là chuyển vị các phần tử đang ghi.



```

1  2  6  7 15
3  5  8 14 16
4  9 13 17 22
10 12 18 21 23
11 19 20 24 25

```

Bài giải: xem trang 174

Bài 110: Viết chương trình thực hiện những yêu cầu sau:

- Cấp phát động ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Viết hàm `int sumNeg(int a[], int m)` trả về số các số âm có trong mảng một chiều `a`, kích thước `m`. Sử dụng hàm này để tìm số các số âm của dòng `k` ($0 \leq k < m$, nhập từ bàn phím) trong ma trận trên.



Nhap n, m: 5 4 ↵

```

-7  -3  -4   3
 3   5   5   9
 7  -6   9   3
 9   4   5   7
 5   3  -2   6

```

Nhap dong k: 2
Dong 2 co 1 so am

Bài giải: xem trang 174

Bài 111: Viết chương trình thực hiện những yêu cầu sau:

- Tạo ma trận có kích thước $n \times m$ (n, m nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-10, 10]$, xuất ma trận.
- Hãy dịch trái xoay vòng các phần tử trong ma trận theo hình tròn xoắn ốc từ trái sang phải, từ ngoài vào trong k bước (k nhập từ bàn phím).



Nhap dong, cot: 4 6 ↵

6	-1	-1	8	-8	-1
0	-8	9	5	-3	-3
-10	3	-5	-2	-2	7
5	-10	-1	0	4	10

 Nhap buoc dịch: 2 ↵

-5	3	6	-1	-1	8
5	-10	0	-8	9	-8
-10	-2	-2	-3	5	-1
-1	0	4	10	7	-3

Bài giải: xem trang 176

Bài 112: Cho ma trận vuông bậc n (nhập từ bàn phím), chứa các phần tử là các số thực ngẫu nhiên. Thực hiện các phép biến đổi sơ cấp hàng trên ma trận (phép khử Gauss) để đưa ma trận trở thành ma trận tam giác trên.



Định lý về các phép biến đổi sơ cấp:

- Định thức trên ma trận không đổi nếu thay một hàng trên ma trận bởi một hàng mới bằng hiệu của hàng cũ với một hàng khác của ma trận nhân với một hệ số k bất kỳ.

- Định thức của ma trận đổi dấu nếu hoán vị hai hàng trên ma trận.

Phép khử Gauss: với mỗi phần tử A_{ii} trên đường chéo chính của ma trận:

- Nếu A_{ii} khác 0, khử các phần tử trên cột i nằm dưới A_{ii} bằng cách thay các hàng A_{jk} bởi hiệu: $A_{jk} - A_{ik} * A_{ji}/A_{ii}$

với $j = (i+1) \dots (n-1)$ và $k = i \dots (n-1)$.

- Nếu A_{ii} bằng 0, tìm phần tử A_{ji} đầu tiên khác 0 khác với $j = (i+1) \dots (n-1)$; nếu tìm thấy, hoán vị hàng i với hàng j (và đổi dấu định thức nếu cần tính); nếu không tìm thấy, định thức bằng 0.



Nhap bac cua ma tran: 3 ↵

0.943	0.217	0.075
0.304	0.165	0.564
0.047	0.178	0.682
0.943	0.217	0.075
0.000	0.095	0.540
0.000	0.000	-0.272

Bài giải: xem trang 180

CHUỖI

Bài 113: Nhập một vào chuỗi số nhị phân. Tìm trong đó chuỗi con dài nhất chứa toàn các số 0.



```
Nhap chuoai nhi phan: 100101000010001001001 ↵
Chuoai 0 dai nhat co 4 ky tu
Bat dau tai s[6]
```

Bài giải: xem trang 181

Bài 114: Viết các hàm mô phỏng các hàm của string.h sau: strlen(), strcpy(), strcat(). Thử nghiệm các hàm này.



```
Chuoai 1: [the quick brown fox ] (20)
Chuoai 2: [jumps over the lazy dog] (23)
strcpy( buf, s1 ) roi strcat( buf, s2 ):
[the quick brown fox jumps over the lazy dog] (43)
```

Bài giải: xem trang 182

Bài 115: Viết các hàm mô phỏng các hàm của string.h sau: strcmp(), strchr(), strrchr(). Thử nghiệm các hàm này.



```
Chuoai goc s: [jackdaws love my big sphinx of quartz]
strchr( s, 'm' ) : [my big sphinx of quartz]
strrchr( s, 'o' ) : [of quartz]
Sap xep cac chuoai dung strcmp():
black blue brown
```

Bài giải: xem trang 184

Bài 116: Viết các hàm mô phỏng của các hàm string.h sau: strstr(), strncmp(), strstr(). Thử nghiệm các hàm này.



```
Chuoaikiem tra : hom qua qua noi qua ma qua khong qua
Vi tri tu 'qua':      x   x           x   x           x
Ky tu dau tien cua s = 'cabbage' khong co trong 'abc' la s[5]
```

Bài giải: xem trang 186

Bài 117: Viết các hàm:

- Mô phỏng hàm LEFT (trả về n ký tự bên trái chuỗi) của Microsoft Excel.
- Mô phỏng hàm RIGHT (trả về n ký tự bên phải chuỗi) của Microsoft Excel.



```
Chuoigoc: [Kernighan and Ritchie]
left( str, 9 ) : [Kernighan]
right( str, 7 ) : [Ritchie]
```

Bài giải: xem trang 189

Bài 118: Viết chương trình nhập vào một chuỗi, ngoài các ký tự chữ cái còn cho phép các ký tự space, tab, nhảy đơn, nhảy đôi, chấm hỏi, chấm than, chấm, phẩy. Xử lý chuỗi như sau:

- Bỏ các ký tự space thừa (các ký tự space đầu chuỗi, cuối chuỗi, giữa các từ chỉ chừa lại một ký tự space).
- Xuất các từ phân biệt, có viết hoa các ký tự đầu mỗi từ, viết thường các ký tự còn lại của từ.



```
Chuoi goc      : [ 'bJARne? sTROUstRUP' ]
Loai space du: ['bJARne? sTROUstRUP']
Cac tu da chuan hoa:
    Bjarne, Stroustrup
```

Bài giải: xem trang 190

Bài 119: Viết chương trình nhập vào một chuỗi tối đa 255 ký tự.

- Đếm số từ có trong chuỗi biết giữa các từ có ít nhất một ký tự space, các dấu “.” và “,”, không có ký tự đặc biệt khác.
- Thống kê tần số xuất hiện các từ có 1, 2, 3, 4, 5, 6, 7 ký tự có trong chuỗi nhập trên.



```
Nhap chuoi: Chieu nay, tren song, khong mot con do. ↵
Co 8 tu
Tan so xuat hien cac tu:
1[0] 2[1] 3[3] 4[2] 5[2] 6[0] 7[0]
```

Bài giải: xem trang 191

Bài 120: Viết chương trình yêu cầu nhập vào hai chuỗi ký tự, mỗi chuỗi có chiều dài tối đa 80 ký tự. Chèn chuỗi 2 vào chuỗi 1 tại vị trí thứ k (k nhập từ bàn phím, với $0 \leq k \leq$ chiều dài chuỗi 1).



```
Chuoi goc      : Em noi sao, roi sao? ↵
Chuoi chen      : doi tre muon ↵
Vi tri chen: 11 ↵
Chuoi ket qua: Em noi sao, doi tre muon roi sao?
```

Bài giải: xem trang 192

Bài 121: Viết chương trình yêu cầu nhập vào chuỗi ký tự có chiều dài tối đa là 80 ký tự. Nhập vào hai số nguyên dương n và p, trong chuỗi trên tiến hành xóa n ký tự bắt đầu từ vị trí p.



```
Nhap chuoi: Con dieu roi cho vuc tham buon theo ↵
Nhap vi tri dau: 9 ↵
Nhap so ky tu loai bo: 17 ↵
Chuoi ket qua: Con dieu buon theo
```

Bài giải: xem trang 193

Bài 122: Viết chương trình nhập vào một số nhị phân ở dạng chuỗi ký tự 0 và 1, chuyển số này thành một số nguyên hệ thập phân.



```
Nhap chuoi nhi phan: 100110111101010010100011 ↵
Tri thap phan: 10212515
```

Bài giải: xem trang 194

Bài 123: Viết chương trình chỉ dùng các thao tác trên chuỗi, cộng hai số nhị phân ở dạng chuỗi ký tự, in ra kết quả cũng ở dạng chuỗi ký tự.



```
1101010000110001
11000000111001
-----
10000010001101010
```

Bài giải: xem trang 195

Bài 124: Viết hàm `itos()` nhận vào một số nguyên dương, chuyển số nguyên dương này thành chuỗi ký tự. Viết hàm `stoi()` ngược lại, nhận vào một chuỗi ký tự số mô tả một số nguyên dương, chuyển chuỗi này thành số nguyên dương.



```
Nhap so nguyen duong n: 12345 ↵
itos() -> 12345
stoi() -> 12345
```

Bài giải: xem trang 197

Bài 125: Viết chương trình nhập vào hai số nguyên dương rất lớn lưu dạng chuỗi ký tự, in ra kết quả nhân hai số nguyên dương trên cũng ở dạng chuỗi ký tự. Trình bày giống như bài toán nhân tay. Thực hiện hoàn toàn bằng các thao tác trên chuỗi.



```
So bi nhan: 87654321 ↵
So nhan   : 12345678 ↵
            87654321
          *
            12345678
          -----
            701234568
            613580247
            525925926
            438271605
            350617284
            262962963
            175308642
            87654321
          -----
            1082152022374638
```

Bài giải: xem trang 198

Bài 126: Viết chương trình nhập một chuỗi và một ký tự. Tìm tất cả các vị trí xuất hiện của ký tự `c` trong một chuỗi `s`.



```
Nhap chuoi: Mai sau khoc nhung roi don dinh menh ↵
Tim ky tu nao? o
Vi tri xuat hien: 10 20 24
```

Bài giải: xem trang 200

Bài 127: Không sử dụng các hàm của `string.h`, viết chương trình đảo các từ của một chuỗi cho sẵn hoặc nhập từ bàn phím. Chuỗi có thể có những ký tự đặc biệt.



```
Chuoi goc: [ cam khong duoc do rac ]
Chuoi dao: [ rac do duoc khong cam ]
```

Bài giải: xem trang 201

Bài 128: Nhập một chuỗi ký tự, xuất ra các từ dài nhất trong chuỗi.



```
Nhap chuoi: Troi dat rong nen tinh yeu de lac ↵
Troi[4] rong[4] tinh[4]
```

Bài giải: xem trang 203

Bài 129: Nhập một chuỗi ký tự chứa ít nhất 4 ký tự số. Loại bỏ một số ký tự ra khỏi chuỗi sao cho 4 ký tự số cuối cùng còn lại (theo đúng thứ tự đó) tạo nên số lớn nhất.



```
Nhap chuoi (it nhat 4 chu so): 24d5n4r05f704n652z393 ↵
So lon nhat con lai: 7693
```

Bài giải: xem trang 204

Bài 130: Viết hàm tìm kiếm trong một số chuỗi cho trước, một chuỗi so trùng với chuỗi mẫu. Cho phép chuỗi mẫu dùng ký tự đại diện (wildcard): *. Ký tự *: so trùng với *không* hoặc *nhiều* ký tự *bất kỳ* tại vị trí tương ứng trong chuỗi cho trước.



```
Danh sach cac tu: television menu editions education
Tim [*e*u*]: menu education
Tim [e*i*n]: education
Tim [*e*o*]: television editions education
```

Bài giải: xem trang 206

Bài 131: Viết chương trình nhập vào chuỗi ký tự có chiều dài tối đa 80 ký tự. Không phân biệt chữ hoa, chữ thường, thực hiện các công việc sau:

- Kiểm tra xem chuỗi có đối xứng hay không?
- Nhập vào một chuỗi ký tự rồi tiến hành xóa hết các ký tự giống với ký tự đó trong chuỗi trên.



```
Nhap chuoi: Stressed? No tips? Spit on desserts! ↵
Chuoi doi xung
Xoa ky tu nao? s ↵
treed? No tip? pit on deert!
```

Bài giải: xem trang 207

Bài 132: Viết chương trình nhập hai chuỗi và xác định xem chúng có được tạo ra với cùng các ký tự hay không. Ví dụ “dear” và “reader” là hai chuỗi tạo từ cùng các ký tự a, d, e, r.



```
Nhap chuoi a: dear ↵
Nhap chuoi b: reader ↵
Tao tu cung cac ky tu
```

Bài giải: xem trang 209

Bài 133: Viết hàm hextoulong() nhận một chuỗi mô tả số unsigned long int thuộc hệ thập lục phân, trả về số unsigned long int thuộc hệ thập phân tương ứng.



Dùng phương pháp Horner¹³ như sau:

- Đặt $x = 0$
- Đặt i bằng thứ tự ký tự trái nhất của chuỗi hex, thứ tự tính từ 0, phải sang trái. h_i là trị hex tại vị trí i .
- Nhân x với 16 rồi cộng h_i với x
- Giảm i đi 1.

Ví dụ: số 0F4D

i	h_i	$x = 16 * x + h$
3	0	0
2	F	$16 * 0 + F = 15$
1	4	$16 * 15 + 4 = 244$
0	D	$16 * 244 + D = 3917$

¹³ William George Horner (1786 - 1837)

5. Nếu $i \geq 0$, lặp lại bước 3, 4
6. Trả về x



Nhap chuoi hex: **3AdE68b1** ↵
 Decimal: 987654321

Bài giải: xem trang 209

Bài 134: Viết hàm `rtoi()` nhận một số La mã dưới dạng chuỗi ký tự, trả về số nguyên tương ứng.



M: 1000 D: 500 C: 100
 L: 50 X: 10 V: 5 I: 1
 Ký hiệu thể hiện 10^x không được đứng ngay trước ký hiệu lớn hơn 10^{x+1} .
 Ví dụ: số 99 là **XCIX**, không phải **IC**.



MCMXCIX = **1999**
 MCMIC = So khong hop le

Bài giải: xem trang 210

Bài 135: Viết chương trình thay thế tất cả các chuỗi con cho trước (thường gọi là các mẫu - pattern) trong một chuỗi bằng một chuỗi khác. Xuất chuỗi kết quả.



Chuoi goc: ta mo thay em o mot noi em xa lam
 Thay the 'em' voi 'em yeu'
 Chuoi moi: Ta mo thay em yeu o mot noi em yeu xa lam

Bài giải: xem trang 211

Bài 136: Nhập vào số dòng văn bản cho đến khi nhấn Ctrl+Z. Viết chương trình đếm số và in số dòng, số từ, tần số, các chữ cái trong các dòng văn bản đó.



Que nha me co gian thien ly, ↵
 Mot khoang ram hien giữa nang trua. ↵
 Va nhung chuyen nghe hoai khong biet chan, ↵
 Bat dau la: Ngay xua, ngay xua... ↵
 ^Z ↵
 4 hang, 29 tu, voi tan so cac ky tu:

A: 17	B: 2	C: 3	D: 1	E: 7	F: 0	G: 9
H: 10	I: 6	J: 0	K: 2	L: 2	M: 3	N: 15
O: 5	P: 0	Q: 1	R: 2	S: 0	T: 5	U: 8
V: 1	W: 0	X: 2	Y: 4	Z: 0		

Bài giải: xem trang 212

Bài 137: Viết chương trình đọc các tên đầy đủ, giữa các từ của tên chỉ có ký tự space, mỗi tên trên một dòng, cho đến khi nhấn Ctrl+Z (hoặc Ctrl+D nếu trên Unix). Sau đó in chúng ta theo dạng niên giám điện thoại chuẩn. Ví dụ:

Wolfgang Amadeus Mozart ⇒ Mozart, Wolfgang A.



George Frederic Handel ↵
 Carl Philipp Emanuel Bach ↵
 Wolfgang Amadeus Mozart ↵
 ^Z ↵
 Handel, George F.

Bach, Carl P. E.
Mozart, Wolfgang A.

Bài giải: xem trang 213

Bài 138: Viết chương trình đọc vào một số dòng, mỗi dòng không quá 50 ký tự, cho đến khi nhấn Ctrl+Z. Sau đó in lại các dòng đó theo dạng thức canh phải với chiều ngang dòng 50 ký tự.



```
Xa em, gio it, lanh nhieu, ↵
Lua khuya tan cham, mua chieu do nhanh. ↵
(Tran Huyen Tran) ↵
^Z ↵

                                Xa em, gio it, lanh nhieu,
                                Lua khuya tan cham, mua chieu do nhanh.
                                (Tran Huyen Tran)
```

Bài giải: xem trang 214

Bài 139: Nén run-length: một chuỗi ký tự (không chứa ký tự số) có thể nén bằng cách sau: Chuỗi con gồm n ($n > 1$) ký tự a giống nhau sẽ được thay thế bằng na (chú ý, n có thể có nhiều hơn 1 ký tự). Ví dụ: `aabcccd` được nén thành `2ab3cd`. Viết chương trình nén và giải nén run-length chuỗi ký tự (không chứa ký tự số) nhập vào.



```
Chuoi goc: aaabccccdddeeeeeeeeeefghhhhhiiiiiaaaabbbbbc
Nen       : 3ab4c4d12efg6h4i4a6bc [44.7%]
Giai nen  : aaabccccdddeeeeeeeeeefghhhhhiiiiiaaaabbbbbc
```

Bài giải: xem trang 215

Bài 140: ISBN (International Standard Book Number, phát âm “is-ben”) là mã số duy nhất cho sách xuất bản trên thế giới. ISBN bao gồm 13 ký tự (0 - 9, ký tự nối - (hyphen), X) chia thành 4 phần bởi ký tự nối: định danh nhóm (quốc gia, ngôn ngữ, ...) định danh nhà xuất bản, định danh sách của nhà xuất bản đó, và số kiểm tra. Số kiểm tra được dùng kết hợp với các số khác trong một thuật toán kiểm tra số ISBN, chỉ chứa một ký tự (0 đến 9, X thay cho 10).

Viết chương trình kiểm tra một số chuỗi ISBN nhập vào có hợp lệ hay không.



Thuật toán đơn giản dùng để kiểm tra tính hợp lệ của số ISBN: lấy từng số của ISBN nhân với số thứ tự chỉ vị trí của nó (bắt đầu từ 1, tính từ phải sang trái, không tính dấu nối -). Tổng các tích nhận được nếu chia hết cho 11 thì số ISBN được kiểm tra là hợp lệ. Ví dụ:

```
ISBN  0 - 1   3   1 - 1   0   3   7   0 - 9
Vị trí 10   9   8   7   6   5   4   3   2   1
Tích   0 + 9 + 24 + 7 + 6 + 0 + 12 + 21 + 0 + 9 = 88
88 chia hết cho 11, vậy số ISBN trên hợp lệ.
```



```
ISBN 0-133-62658-x hợp lệ
```

Bài giải: xem trang 216

ĐỀ QUY

Bài 141: Giả sử dân số thế giới năm 2000 là 8 tỷ người với mức tăng trưởng hàng năm không đổi là 2,5%. Viết hàm đệ quy tính dân số thế giới năm 2010.



```
10240676354 nguoi
```

Bài giải: xem trang 217

Bài 142: Viết chương trình nhập vào một dòng văn bản, kết thúc bằng phím Enter, sau đó hiển thị dòng văn bản đảo của văn bản nhập. Dùng giải thuật đệ quy với hàm main(), không dùng các cấu trúc lưu trữ như mảng, chuỗi.



Ngôn ngữ C cho phép đệ quy ngay cả với hàm main()



```
race fast safe car ↵
rac efas tsaf ecar
```

Bài giải: xem trang 218

Bài 143: Cho hàm Ackermann¹⁴ với n và m không âm:

$$A(n, m) = \begin{cases} m + 1 & n = 0 \\ A(n - 1, 1) & m = 0 \\ A(n - 1, A(n, m - 1)) & n, m > 0 \end{cases}$$

Tính $A(3, 6)$. Cho biết số lần gọi đệ quy khi tính $A(3, 6)$.



```
A( 3, 6 ) = 509
Goi de quy 172233 lan
```

Bài giải: xem trang 219

Bài 144: Viết hàm đệ quy double Pi() không tham số tính số Pi theo công thức:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^k \frac{1}{2k+1} \quad (\text{với sai số epsilon } 10^{-3}, 4/n < \text{epsilon})$$



```
Pi = 3.141
```

Bài giải: xem trang 220

Bài 145: Viết các hàm đệ quy

void ToBin(int n) \Rightarrow xuất dạng nhị phân của n (n nguyên dương).

void ToHex(int n) \Rightarrow xuất dạng thập lục phân của n (n nguyên dương).



```
Nhap x: 123 ↵
Bin: 1111011
Hex: 7B
```

Bài giải: xem trang 221

Bài 146: Nhập vào số nguyên n ($0 < n \leq 9$). Xuất tam giác Pascal chiều cao n .

¹⁴ Wilhelm Ackermann (1896 - 1962)



Tam giác Pascal¹⁵ được xác định như sau:

$$C_0^0$$

$$C_1^0 \quad C_1^1$$

...

$$C_n^0 \quad C_n^1 \dots C_n^n$$

Trong đó $C_n^k = \frac{n!}{k!(n-k)!}$, $0 \leq k \leq n$ là tổ hợp n chập k (trong trường hợp này gọi là các hệ số của nhị thức Newton¹⁶).

C_n^k còn được xác định bằng công thức truy hồi:

$$C_n^k = \begin{cases} 1 & k = 0 \\ 1 & k = n \\ C_{n-1}^k + C_{n-1}^{k-1} & 0 < k < n \end{cases}$$



Nhap n (n < 9): 4 ↵

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
  
```

Bài giải: xem trang 222

Bài 147: Viết hàm sắp xếp *tăng* một mảng theo kiểu chọn (SelectionSort) với giải thuật đệ quy: void SelectSort(int a[], int n) (n là số phần tử).



```

Mang goc : 3 5 4 6 7 1 2
Mang tang: 1 2 3 4 5 6 7
  
```

Bài giải: xem trang 223

Bài 148: Viết hàm sắp xếp *giảm* một mảng theo kiểu nổi bọt (BubbleSort) với giải thuật đệ quy: void BubbleSort(int a[], int n) (n là số phần tử).



```

Mang goc : 3 5 4 6 7 1 2
Mang giam: 7 6 5 4 3 2 1
  
```

Bài giải: xem trang 224

Bài 149: Dùng giải thuật đệ quy, viết hàm tìm kiếm nhị phân trên mảng đã được sắp xếp: int BSearch(int a[], int x, int left, int right) (left là biên trái, right là biên phải của mảng, x là số cần tìm trong mảng).



```

2 3 4 5 6 7
Nhap x: 5 ↵
a[3]
  
```

Bài giải: xem trang 225

¹⁵ Blaise Pascal (1623 - 1662)

¹⁶ Isaac Newton (1643 - 1727)

Bài 150: Dùng giải thuật đệ quy, viết hàm `int OddSum(int a[], int n)` trả về tổng các phần tử có trị lẻ trong mảng.



```
2 3 4 5 6 7
Tong cac phan tu le: 15
```

Bài giải: xem trang 227

Bài 151: Dùng giải thuật đệ quy, viết hàm `int MaxArr(int a[], int n)` trả về vị trí của phần tử có trị lớn nhất trong một mảng không rỗng.



```
-2 3 -4 -5 6 -8
Max = a[4] = 6
```

Bài giải: xem trang 228

Bài 152: Dùng giải thuật đệ quy, viết hàm `int isSym(int a[], int left, int right)` kiểm tra xem một mảng có đối xứng hay không (left là biên trái, right là biên phải, trả về 1 nếu mảng đối xứng).



```
7 -2 3 4 3 -2 7
Mang doi xung
```

Bài giải: xem trang 229

Bài 153: Dùng giải thuật đệ quy, viết hàm `float NegAverage(int a[], int n)` trả về trị trung bình các phần tử âm của một mảng chứa các số nguyên.



```
-2 3 -4 -5 6 -8
Trung binh cong cac phan tu am: -4.75
```

Bài giải: xem trang 230

Bài 154: Dùng giải thuật đệ quy, viết chương trình `FACTORIAL.C`, tính giai thừa các số nguyên dương. Các số nguyên dương được nhập vào từ *tham số dòng lệnh*.



```
FACTORIAL 9 3A 12 ↵
9! = 362880
3A: so khong hop le
12! = 479001600
```

Bài giải: xem trang 231

Bài 155: Viết hàm đệ quy $F(n)$ tính số Fibonacci thứ n (n nguyên, $0 < n < 40$, nhập từ bàn phím) theo công thức truy hồi:

$$F_n = \begin{cases} 1 & n = 1, 2 \\ F_{n-1} + F_{n-2} & n > 2 \end{cases}$$

Kiểm tra lại bằng công thức dạng đóng (của A. de Moivre) dùng tính số Fibonacci thứ n như sau: $F_n = \frac{\varphi^n - \psi^n}{\sqrt{5}}$ với $\varphi = \frac{1 + \sqrt{5}}{2}$, $\psi = 1 - \varphi$ (φ gọi là golden mean).



```
Nhap n (0 < n < 40): 24 ↵
De quy : Fi(24) = 46368
Cong thuc dong: Fi(24) = 46368
```

Bài giải: xem trang 231

Bài 156: Viết hàm đệ quy tính phân số liên tục sau:

$$F = 1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\dots + \frac{1}{101 + \frac{1}{103}}}}}$$



F = 1.31304

Bài giải: xem trang 232

Bài 157: Viết giải thuật đệ quy, tính định thức của một ma trận vuông bậc n (nhập từ bàn phím), trị các phần tử của ma trận lấy ngẫu nhiên trong đoạn $[-10, 10]$.



Công thức truy hồi dùng tính định thức ma trận A vuông bậc n:

$$\det(A) = \begin{cases} a_{11} & n = 1 \\ \sum_{j=1}^n (-1)^{1+j} a_{1j} \det(A_{1j}) & n > 1 \end{cases}$$

Với (A_{1j}) là ma trận con vuông bậc n-1, tạo từ ma trận A bỏ đi các phần tử trên dòng 1, cột j (dòng và cột của phần tử a_{1j})



Nhap bac ma tran: 4 ↵

```
10  -2  3  9
 7   6  2 -10
 9  10  4  -5
 3  -4 -3  -2
```

Det(A) = 2448

Bài giải: xem trang 233

Bài 158: Dùng giải thuật đệ quy, viết chương trình GCD (Greatest Common Divisor), tính ước số chung lớn nhất của 2 số nguyên thập vào từ bàn phím. Chương trình phải xử lý các trường hợp nhập số 0 hoặc số âm.



Thuật toán Euclid¹⁷ với công thức truy hồi:

$$\gcd(a, b) = \begin{cases} b & a = 0 \\ \gcd(b \bmod a, a) & a > 0 \end{cases}$$



Nhap hai so: 0 0 ↵

GCD (0, 0): khong xac dinh

Nhap hai so: 0 3 ↵

GCD (0, 3) = 3

Nhap hai so: -18 27 ↵

GCD(-18, 27) = 9

Bài giải: xem trang 235

¹⁷ Euclid of Alexandria (325 BC - 265 BC)

Bài 159: Một palindrome là một từ hay một câu, đọc xuôi ngược đều giống như nhau. Viết chương trình, với giải thuật đệ quy, đọc từng dòng từ bàn phím vào và báo cho biết đó có phải là palindrome không. Với mỗi dòng, bỏ qua các ký tự không phải alphabet và không quan tâm đến chữ hoa, chữ thường.



```
O give me a home ↵
-> không phải palindrom
Madam, I'm Adam! ↵
-> Palindrome
^Z ↵
```

Bài giải: xem trang 236

Bài 160: Viết hàm đệ quy in ra tất cả $n!$ các hoán vị của chuỗi n ký tự đầu bảng alphabet. Áp dụng tìm tất cả các hoán vị của chuỗi ABC ($n = 3$).



```
Nhap n: 3 ↵
ABC ACB BAC BCA CBA CAB
```

Bài giải: xem trang 236

Bài 161: Viết các hàm đệ quy tính $\sin(x)$ và $\cos(x)$ theo cặp đồng nhất thức:

$$\begin{cases} \sin 3x = (4 \cos^2 x - 1) \sin x \\ \cos 3x = (1 - 4 \sin^2 x) \cos x \end{cases}$$

và cặp đồng nhất thức xấp xỉ cho các giá trị rất nhỏ của x :

$$\begin{cases} \sin x \approx x - x^3 / 6 \\ \cos x \approx 1 - x^2 / 2 \end{cases}$$

Với độ chính xác $5 \cdot 10^{-4}$. Kiểm tra với các hàm tương ứng của `math.h`.



```
Doi chung tinh bang math.h trong ()
sin(-2.1) = -0.8632093666 ( -0.8632093666 )
cos(-2.1) = -0.5048461046 ( -0.5048461046 )
```

Bài giải: xem trang 237

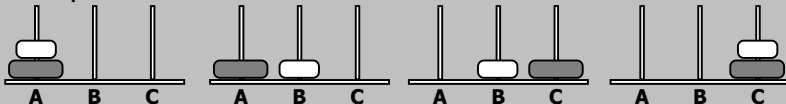
Bài 162: In các lần di chuyển đĩa trong bài toán tháp Hanoi với 4 đĩa.



Bài toán tháp Hanoi (Towers of Hanoi): có 3 cọc A, B, C; khởi đầu cọc A có n đĩa xếp sao cho đĩa lớn hơn luôn nằm bên dưới, 2 cọc kia trống.

Hãy chuyển tất cả đĩa từ cọc A sang cọc C, được dùng cọc phụ B. trong quá trình chuyển đĩa phải đảm bảo đĩa lớn hơn luôn nằm bên dưới.

Minh họa với 2 đĩa:



Xem thêm về bài này trong nhiều sách về lập trình.



```
Disk 1: [1] -> [2]
Disk 2: [1] -> [3]
Disk 1: [2] -> [3]
Disk 3: [1] -> [2]
Disk 1: [3] -> [1]
```

```

Disk 2: [3] -> [2]
Disk 1: [1] -> [2]
Disk 4: [1] -> [3]
Disk 1: [2] -> [3]
Disk 2: [2] -> [1]
Disk 1: [3] -> [1]
Disk 3: [2] -> [3]
Disk 1: [1] -> [2]
Disk 2: [1] -> [3]
Disk 1: [2] -> [3]

```

Bài giải: xem trang 238

Bài 163: Có thể tính x^n chỉ cần khoảng $\log_2 n$ phép nhân dựa vào nhận xét:

$$x^n = \begin{cases} 1 & n = 0 \\ (x^k)(x^k) & n = 2k \quad k \in \mathbb{N} \\ x(x^k)(x^k) & n = 2k + 1 \end{cases}$$

Hãy tính x^n với x là số thực, n là số nguyên, nhập từ bàn phím. Kiểm tra đối chứng bằng hàm `pow()` của `math.h`.



```

Nhap x, n: 8.5 - 2 ↵
mypow() : 0.013841
pow() : 0.013841

```

Bài giải: xem trang 239

STRUCTURE - UNION - BIT FIELD

Bài 164: Viết chương trình cài đặt cấu trúc của một số phức và các hàm thực hiện các phép toán trên nó.



Nếu $z = a + bi$ và $w = c + di$ là hai số phức, thì:

$$z + w = (a + c) + (b + d)i$$

$$z - w = (a - c) + (b - d)i$$

$$zw = (ac - bd) + (ad + bc)i$$

$$\frac{z}{w} = \left(\frac{ac + bd}{c^2 + d^2} \right) + \left(\frac{bc - ad}{c^2 + d^2} \right)i$$



```

Nhap mot so phuc:
  Phan thuc: 3.5 ↵
  Phan ao : 0.7 ↵
Nhap mot so phuc:
  Phan thuc: 4.2 ↵
  Phan ao : 2.6 ↵
a + b = 7.7 + 3.3i
a - b = -0.7 - 1.9i
a * b = 12.9 + 12.0i
a / b = 0.7 - 0.3i

```

Bài giải: xem trang 240

Bài 165: Viết chương trình cài đặt cấu trúc của một phân số và các hàm thực hiện các phép toán số học trên nó.



```
Nhap tu so va mau so: 1 -2 ↵
Nhap tu so va mau so: -3 4 ↵
a + b = -5/4
a - b = 1/4
a * b = 3/8
a / b = 2/3
```

Bài giải: xem trang 242

Bài 166: Viết chương trình cài đặt cấu trúc lưu thông tin của đường tròn, bao gồm tọa độ tâm và bán kính của đường tròn. Nhập dữ liệu của hai đường tròn, xác định vị trí tương đối giữa chúng.



Với mỗi cặp đường tròn, xét 3 trường hợp:

- Tâm đường tròn này nằm ngoài đường tròn kia.
- Hai tâm trùng nhau.
- Tâm đường tròn này nằm trên hoặc trong đường tròn kia (trừ tâm).

Dựa vào khoảng cách giữa hai tâm với bán kính của mỗi đường tròn để xác định vị trí tương đối giữa chúng.



```
Nhap xc, yc va R cua C1: 2.23 1.12 1.67 ↵
Nhap xc, yc va R cua C2: 1.32 2.41 3.25 ↵
Trong nhau
```

Bài giải: xem trang 243

Bài 167: Viết chương trình cài đặt cấu trúc một đơn thức (một tham số), và dùng mảng các cấu trúc này để lưu một đa thức. Nhập vào hai đa thức, thực hiện phép toán nhân hai đa thức này; sau đó tính trị của đa thức kết quả với trị của tham số nhập vào từ bàn phím.



Tích của 2 đa thức 1 tham số x , $f(x)$ bậc m và $g(x)$ bậc n là:

$$f(x) \cdot g(x) = c_{m+n}x^{m+n} + c_{m+n-1}x^{m+n-1} + \dots + c_1x + c_0$$

Trong đó c_k bằng tổng các tích $a_i b_j$ mà $i + j = k$ ($k = 0, 1, \dots, m + n$)



Tính trị đa thức tổng quát bậc n : $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ phải thực hiện $n(n+1)/2$ phép nhân (x^k tính $k-1$ phép nhân). Nếu tính trị của đa thức bằng phương pháp Horner sau đây, chỉ cần thực hiện n phép nhân:

$$P(x) = ((\dots((a_n x + a_{n-1})x + a_{n-2})x + \dots + a_2)x + a_1)x + a_0$$


```
Nhap bac da thuc: 3 ↵
Nhap 4 he so: 5 -4 7 -2
Nhap bac da thuc: 2 ↵
Nhap 3 he so: 3 -2 7 ↵
p1(x) = 5x^3-4x^2+7x-2
p2(x) = 3x^2-2x+7
Da thuc ket qua:
p(x) = 15x^5-22x^4+64x^3-48x^2+53x-14
Nhap x: 1.2 ↵
p(1.2) = 82.7776
```

Bài giải: xem trang 244

Bài 168: Viết chương trình cài đặt một mảng các cấu trúc lưu trữ thông tin sách trong thư viện, bao gồm: tựa sách, ISBN, tên tác giả, tên nhà xuất bản, ngày tháng năm nhập sách (là ngày viết phiếu). Sau đó, nhập vào một chuỗi ISBN, tìm và in ra thông tin sách tương ứng nếu có.



Dùng struct tm của time.h để lưu trữ ngày cập nhật phiếu.



```
Nhap thong tin sach:
Tua      > The C Programming Language ↵
ISBN     > 0-131-10370-9 ↵
Tac gia  > Kernighan, Brian W. & Ritchie, Dennis M. ↵
NXB      > Prentice Hall ↵
Tiep ( y/n )? y ↵
Nhap thong tin sach:
Tua      > Applications Programming in ANSI C ↵
ISBN     > 0-023-61141-3 ↵
Tac gia  > Johnsonbaugh, R. & Kalin M. ↵
NXB      > MacMillan Pub. Co. ↵
Tiep ( y/n )? n ↵
ISBN ? 0-023-61141-3 ↵
Ket qua tim:
Applications Programming in ANSI C
Johnsonbaugh, R. & Kalin M.
MacMillan Pub. Co.
[update: 15-04-2006]
```

Bài giải: xem trang 246

Bài 169: Viết chương trình cài đặt một cấu trúc lưu thông tin chấm công ngày (time card) của một tổ 6 công nhân, bao gồm: mã số công nhân, giờ vào làm và giờ nghỉ. Một đợt vào làm không được làm vượt quá 2 ca liên tiếp. In bảng thanh toán công ngày của tổ công nhân đó. Biết:

Ca đêm	01 giờ - 06 giờ, 18 giờ - 24 giờ	15000đ/giờ
Ca ngày	06 giờ - 18 giờ	10000đ/giờ



```
Nhap ID cong nhan 1: 78-125 ↵
Vao (gio phut): 1 45 ↵
Ra (gio phut) : 5 30 ↵
...
Nhap ID cong nhan 6: 89-273 ↵
Vao (gio phut): 21 25 ↵
Ra (gio phut) : 3 40 ↵
78-125 01:45 05:30 56250
45-224 03:10 09:50 80833
52-436 06:20 01:30 Nhap sai!
82-729 14:40 20:30 70833
67-451 18:45 23:10 70000
```

Bài giải: xem trang 248

Bài 170: Viết chương trình đặt một cấu trúc lưu 52 lá bài của bộ bài tây, bao gồm nước (face: ách, hai, ba, ..., mười, bồi, đầm, già) và chất (suit: cơ, rô, chuồn, bích). Thực hiện việc chia bài ngẫu nhiên cho 4 người chơi.



Hãy trộn bài (shuffle) *ngẫu nhiên*, trước khi chia theo *thứ tự*.



```
Chín Rô      Bon Co      Bay Bích     Dam Rô
Sau Co       Bay Rô      Nam Rô       Ba Rô
...
Già Chuồn   Boi Rô      Nam Chuồn   Chín Bích
Bon Chuồn   Sau Rô      Ach Rô      Sau Bích
```

Bài giải: xem trang 250

Bài 171: Viết chương trình nhập vào một ký tự. Không dùng các toán tử thao tác trên bit (bitwise operators) cũng như không thực hiện phép chia cho 2, hãy in dạng hiển thị nhị phân của ký tự này.



Dùng union kết hợp với bit field.



```
Nhap 1 ky tu: Z ↵
0 1 0 1 1 0 1 0
```

Bài giải: xem trang 251

TẬP TIN

Bài 172: Viết chương trình FILESIZE.C dùng xác định kích thước của một số tập tin. Tên các tập tin cần được xác định kích thước được nhập như là tham số dòng lệnh của FILESIZE.



```
FILESIZE readme.txt list.dat ↵
readme.txt [2148 byte(s)]
list.dat [1246 byte(s)]
```

Bài giải: xem trang 252

Bài 173: Viết chương trình ghi 5 số nguyên ngẫu nhiên vào tập tin INTERGER.DAT, 5 số thực ngẫu nhiên vào tập tin REAL.DAT. Sau đó đọc các số này từ tập tin và xuất ra màn hình.



```
Ghi xong file...
28764    13997    19450    7297    24082
0.14554  0.13376  0.82284  0.35276  0.79724
Doc xong file...
```

Bài giải: xem trang 253

Bài 174: Tập tin văn bản PERSON.DAT lưu thông tin cá nhân thành các dòng có định dạng như sau (chú thích trong () là yêu cầu của trường):

code(unsigned int):firstname lastname(32),address(32):birthday(mm/dd/yy)

Viết chương trình đọc tập tin PERSON.DAT, lấy và hiển thị thông tin lưu trữ ứng với từng cá nhân.



```
Nội dung tập tin PERSON.DAT
1654:Jackie Chan,Hong Kong:7/22/54
4424:Tony Jaa,Thailand:5/12/76
Kết quả
Jackie Chan                                [code: 1654]
  Address : [Hong Kong]
  Birthday: [7/22/54]
Tony Jaa                                    [code: 4424]
  Address : [Thailand]
  Birthday: [5/12/76]
```

Bài giải: xem trang 254

Bài 175: Viết chương trình UPPER.C khi chạy sẽ chuyển đổi tất cả ký tự thường của một tập tin (nhập tên tập tin từ dòng lệnh) thành ký tự hoa chứa vào một tập tin khác (nhập tên tập tin từ dòng lệnh).



```
UPPER lower.txt upper.txt ↵
Chuyen thanh chu hoa xong...
```

Bài giải: xem trang 256

Bài 176: Viết chương trình CIPHER.C có hai chức năng: mã hóa nội dung một tập tin và giải mã nội dung tập tin được mã hóa. Luật mã hóa: ký tự nào có mã ASCII nhỏ hơn 128 thì chuyển mã thành: mã ký tự đó + 128; ký tự nào có mã ASCII lớn hơn 128 thì chuyển mã thành: mã ký tự đó - 128.



```
Mã hóa:
CIPHER origin.txt encode.txt ↵
Xu ly xong...
Giải mã:
CIPHER encode.txt origin.txt ↵
Xu ly xong...
```

Bài giải: xem trang 257

Bài 177: Viết chương trình TESTFILE.C sử dụng theo cú pháp sau:

TESTFILE num filename : (*number*) tạo tập tin filename chứa num ký tự ngẫu nhiên viết hoa.

TESTFILE -v filename : (*view*) in nội dung tập tin filename ra màn hình.

TESTFILE -r filename : (*reverse*) in ngược nội dung tập tin filename ra màn hình.



```
TESTFILE 10 test.txt ↵
File da duoc tao...
TESTFILE -v test.txt ↵
F L P C Q P Z D L O
Cuoi file...
TESTFILE -r test.txt ↵
O L D Z P Q C P L F
Dau file...
```

Bài giải: xem trang 258

Bài 178: Viết chương trình LINES.C hiển thị nội dung của tập tin nguồn với từng dòng được đánh số tại đầu dòng. Cú pháp sử dụng:

LINES < SOURCE.C : hiển thị tập tin SOURCE.C có đánh số dòng.

LINES < SOURCE.C > NSOURCE.C: tập tin mới NSOURCE.C sẽ có đánh số dòng.



```
LINES < SOURCE.C ↵
0: #include <stdio.h>
1:
2: int main ()
3: {
...
17: return 0;
18: }
```

Bài giải: xem trang 259

Bài 179: Viết chương trình thực hiện các tác vụ sau:

- Tạo tập tin chứa các mẫu tin theo cấu trúc sau:

```
struct STUDENT {
    int code;           /* mã số */
    char name [20];     /* tên */
    double avgmark;     /* điểm trung bình */
};
```

- Nhập thông tin một số sinh viên, thêm vào tập tin (giả sử nhập trường code không trùng, không cần kiểm tra).

- In tất cả các mẫu tin lưu trong tập tin.

- Nhập code, tìm mẫu tin trong tập tin theo code; nếu tìm được, cho phép cập nhật lại trường avgmark của mẫu tin đó.

- Dùng một menu đơn giản để hiển thị các tác vụ.



```
Ten file? STUDENT.DAT ↵
MENU (File 'STUDENT.DAT')
----
[1]. Them
[2]. Doc
[3]. Sua
[4]. Thoat
Chon tac vu: 1 ↵
GHI FILE
Nhap mot mau tin (y/n)? y ↵
Ma so > 7366 ↵
Ten > Steve Blenheim ↵
Diem TB > 7.84 ↵
Nhap mot mau tin (y/n)? y ↵
Ma so > 8376 ↵
Ten > James Ikeda ↵
Diem TB > 8.21 ↵
Nhap mot mau tin (y/n)? n ↵
Da ghi file...

MENU (File 'STUDENT.DAT')
----
[1]. Them
[2]. Doc
[3]. Sua
[4]. Thoat
```

```

Chon tac vu: 2 ↵
DOC FILE
1 7366 Steve Blenheim 7.84
2 8376 James Ikeda 8.21
Tong cong: 2 record(s)

MENU (File 'STUDENT.DAT')
----
[1]. Them
[2]. Doc
[3]. Sua
[4]. Thoat
Chon tac vu: 3 ↵
SUA
Ma so > 8376 ↵
James Ikeda
Diem moi > 7.94 ↵
Da cap nhat...

MENU (File 'STUDENT.DAT')
----
[1]. Them
[2]. Doc
[3]. Sua
[4]. Thoat
Chon tac vu: 4 ↵
Bye...

```

Bài giải: xem trang 260

Bài 180: Viết chương trình đọc một tập tin văn bản. Sau đó trình bày những thống kê sau: số dòng, số từ, tần số xuất hiện của các từ trong tập tin đó.



```

Nội dung tập tin HANOI.TXT
Ta con em, cay bang mo coi mua Dong...
Goc pho mo coi mua Dong, manh trang mo coi mua Dong
Mua Dong nam ay...
Kết quả:
Nhap ten file: HANOI.TXT ↵
File HANOI.TXT co 3 dong, 25 tu, voi tan so cac tu:
    ta: 1      con: 1      em: 1      cay: 1
    bang: 1    mo: 3      coi: 3      mua: 4
    dong: 4    goc: 1      pho: 1      manh: 1
    trang: 1   nam: 1      ay: 1

```

Bài giải: xem trang 263

Bài 181: Viết chương trình nhận vào chuỗi s, sau đó đọc một tập tin văn bản. In số thứ tự các dòng có chứa chuỗi s; với mỗi dòng đó in các vị trí có xuất hiện chuỗi s.



```

Nội dung tập tin TONGBIET.TXT
Dua nguoi, ta khong dua sang song,
Sao lai thay song o trong long?
Bong chieu khong tham, khong vang vot
Sao day hoang hon trong mat trong?
Kết quả:
Nhap ten file: THO.TXT ↵

```

```
Nhap chuoi tim: khong ↵
Dong 0: 14
Dong 2: 11 23
```

Bài giải: xem trang 265

Bài 182: Viết chương trình tạo một tập tin nhị phân chứa các mẫu tin, mỗi mẫu tin bao gồm tên, số điện thoại, địa chỉ của một người. Sau đó đọc tập tin nhị phân này và tạo một tập tin văn bản chứa kết quả đọc được, trình bày ở dạng bảng.



```
Tao file nhi phan PERSON.DAT...
Nhap mot mau tin (y/n)? y ↵
Ten > Popeye Sailor ↵
Dien thoai > (156)454-3325 ↵
Dia chi > 94 Comics Str., Anywhere ↵
Nhap mot mau tin (y/n)? y ↵
Ten > James Bond ↵
Dien thoai > (846)233-2837 ↵
Dia chi > 07 Movies Str., Cinema ↵
Nhap mot mau tin (y/n)? n ↵
Tao file van ban PERSON.TXT...
Hien thi file van ban PERSON.TXT...
Popeye Sailor (156)454-3325 94 Comics Str., Anywhere
James Bond (846)233-2837 07 Movies Str., Cinema
```

Bài giải: xem trang 265

Bài 183: Viết một hàm dùng giải thuật đệ quy đọc các dòng văn bản (không quá 255 ký tự) từ một tập tin. Sau đó ghi các dòng văn bản này theo một thứ tự đảo ngược vào một tập tin khác.



```
REVERSE test.txt reverse.txt ↵
Dao nguc de quy xong...
```

Bài giải: xem trang 267

Bài 184: Tập tin EMP.DAT chứa các hồ sơ nhân viên, mỗi hồ sơ là một cấu trúc bao gồm tên, ngày sinh (là một cấu trúc bao gồm ngày, tháng, năm sinh), lương. Viết chương trình nhập, sắp xếp các mẫu tin trong tập tin trên theo thứ tự tăng của tuổi, xuất danh sách đã sắp xếp. Thao tác trên tập tin, không dùng mảng tạm.



```
Nhap so nhan vien: 3 ↵
Nhap (ten, ngay, thang, nam sinh, luong):
1 > Dong 30 4 1982 350000 ↵
2 > Tay 26 3 1980 150000 ↵
3 > Nam 1 9 1982 400000 ↵
Nhap du lieu xong ...
Xuat danh sach sap xep:
Nam      1/9/1982      400000
Dong     30/4/1982    350000
Tay      26/3/1980    150000
```

Bài giải: xem trang 268

Bài 185: Mỗi ma trận kích thước $m \times n$ được lưu trong một tập tin văn bản theo quy ước sau: dòng đầu chứa trị m (số dòng) và trị n (số cột); m dòng tiếp theo, mỗi dòng lưu n phần tử trên các dòng tương ứng của ma trận, cách nhau bởi dấu space. Viết chương trình nhận các tham số dòng lệnh là tên các tập tin văn bản, thực hiện công việc sau:

- Nếu có hai tham số dòng lệnh: tạo hai ma trận (kích thước nhập từ bàn phím), có các phần tử là các số thực ngẫu nhiên và lưu vào hai tập tin văn bản có tên tương ứng với các tham số dòng lệnh.
- Nếu có ba tham số dòng lệnh: nhân hai ma trận chứa trong các tập tin văn bản có tên là tham số dòng lệnh thứ nhất và thứ hai, sau đó lưu ma trận kết quả vào tập tin văn bản có tên là tham số dòng lệnh thứ ba. Nếu phép nhân ma trận không thực hiện được, không tạo tập tin kết quả.



```
MULMATRIX MATRIX1.TXT MATRIX2.TXT ↵
Nhập m, n: 3 4 ↵
Tạo xong file chứa ma tran...
Nhập m, n: 4 5 ↵
Tạo xong file chứa ma tran...

MULMATRIX MATRIX1.TXT MATRIX2.TXT MATRIX3.TXT ↵
Tạo xong file chứa ma tran...
```

Bài giải: xem trang 270

Bài 186: Thông tin một quyển sách trong thư viện được lưu trong tập tin văn bản BOOKS.TXT thành một khối như sau:

```
C How to Program
Deitel, H.M. & Deitel, P.J.
Prentice Hall, 2001
ISBN 0-13-089572-5
*
```

Các khối thông tin lưu liên tục nhau và cách nhau bởi dòng có dấu * (các dòng khác trong khối không được có dấu * đầu dòng)

Để tìm kiếm nhanh, người ta tạo một tập tin nhị phân INDEX.IDX lưu các mẫu tin gồm tên sách và vị trí của khối thông tin về tên sách đó trong tập tin BOOKS.TXT.

Viết chương trình LOOK sử dụng theo cú pháp sau:

LOOK -i BOOKS.TXT INDEX.IDX : Tạo tập tin INDEX.IDX.

LOOK -v BOOKS.TXT INDEX.IDX : Hỏi tên sách, tìm trong BOOKS.TXT
nhanh nhờ INDEX.IDX, in thông tin.



```
LOOK -i BOOKS.TXT INDEX.IDX ↵
Tạo xong file index...
LOOK -v BOOKS.TXT INDEX.IDX ↵
Nhấn Ctrl+Z để dừng
Tên sách? C How to Program ↵
C How to Program
Deitel, H.M & Deitel, P.J.
Prentice Hall, 2001
ISBN 0-13-089572-5
Tên sách? ^Z ↵
```

Bài giải: xem trang 273

Bài 187: Viết chương trình hiển thị nội dung một tập tin văn bản trên thiết bị xuất chuẩn (giả sử là màn hình văn bản 80 x 25); mỗi lần hiển thị đầy các dòng của màn hình, sau đó dừng lại chờ nhấn phím Enter để hiển thị tiếp.

Bài giải: xem trang 275

Bài 188: Cho hai tập tin văn bản chứa các số nguyên được sắp xếp theo thứ tự tăng dần. Viết một chương trình để trộn (merge) hai tập tin này vào một tập tin số nguyên thứ ba cũng được sắp theo thứ tự tăng dần (không dùng mảng phụ). Tên của cả ba tập tin được nhập vào như các tham số dòng lệnh.



```
Nội dung tập tin NUMBER1.TXT: 1 5 9 13 17 19
Nội dung tập tin NUMBER2.TXT: 2 6 10 14 18 22
SORT NUMBER1.TXT NUMBER2.TXT NUMBER.TXT ↵
Tron ket thuc...
Nội dung tập tin NUMBER.TXT: 1 2 5 6 9 10 13 14 17 18 19 22
```

Bài giải: xem trang 276

Bài 189: Viết chương trình chia (split) một tập tin thành nhiều tập tin nhỏ có kích thước tối đa n. Các tập tin nhỏ này cùng tên với tập tin ban đầu, với phần mở rộng được đánh số.

Chương trình nhận hai tham số dòng lệnh: tên tập tin cần chia và kích thước tối đa n (tính bằng kb).



Kiểm tra kết quả bằng cách nối các tập tin lại (lệnh COPY /B) hoặc viết chương trình kết hợp (combine) các tập tin con đã chia ở trên thành một tập tin duy nhất.



```
Tập tin bigfile.dat có kích thước 58436 bytes
SPLIT bigfile.dat 20 ↵
File bigfile.000 [20480 byte(s)]
File bigfile.001 [20480 byte(s)]
File bigfile.002 [17476 byte(s)]
Chia file ket thuc... [3 file(s)]
```

Bài giải: xem trang 277

Bài 190: Hiển thị nội dung của một tập tin dưới dạng số thập lục phân. Cách trình bày này (gọi là “hex dump”) thường thấy trong các tiện ích xem tập tin phổ biến. Xem ví dụ minh họa.



```
+0 +1 +2 +3 ... +C +D +E +F Contents
00000000 23 69 6E 63 ... 64 69 6F 2E #include <stdio.
00000010 68 3E 0D 0A ... 20 3C 63 74 h> #include <ct
00000020 79 70 65 2E ... 69 7A 65 5F ype.h>      size_
...
00000450 7D 20 0D 0A                                }
1108 bytes
```

Bài giải: xem trang 278

CÁC VẤN ĐỀ KHÁC

Bài 191: Dùng các hàm chuẩn của time.h viết hàm nhập vào tháng và năm (sau 1900). In lịch bằng tiếng Việt (không dấu) của tháng đó chỉ trong 5 dòng.



```
Nhap thang, nam (sau 1900): 7 2006 ↵
Thang Bay 2006
  CN Hai Ba Tu Nam Sau Bay
  30 31
    2 3 4 5 6 7 8
    9 10 11 12 13 14 15
   16 17 18 19 20 21 22
   23 24 25 26 27 28 29
```

Bài giải: xem trang 279

Bài 192: Dùng các hàm chuẩn của time.h, viết chương trình hiển thị thứ, ngày, tháng, năm, giờ hiện tại, múi giờ GMT, bằng tiếng Việt (không dấu).



```
Hom nay Chu Nhat, ngay 09 thang Hai nam 2003
Bay gio la 05:24:18 AM +007 GMT
```

Bài giải: xem trang 281

Bài 193: Tạo menu cho chương trình trong bài 178 (trang 50) bằng cách dùng mảng các con trỏ hàm (functor).



```
MENU (File 'STUDENT.DAT')
----
[1]. Them
[2]. Doc
[3]. Sua
[4]. Thoat
Chon tac vu:
```

Bài giải: xem trang 283

Bài 194: Viết một hàm nhận danh sách tham số gồm: tên một hàm lượng giác (có một tham số double) của math.h, cận dưới và cận trên của miền cần tính trị, bước tăng khi tính trị. In ra bảng các trị của hàm đó trong miền và bước tăng được chọn.



```
Goi ham khi can tinh sin(x) trong doan [0, π], với bước tăng π/8:
tab_func( sin, 0.0, M_PI, M_PI/8.0 );
Kết quả:
```

x	f(x)
0.0000	0.0000
0.3927	0.3827
0.7854	0.7071
1.1781	0.9239
1.5708	1.0000
1.9635	0.9239
2.3562	0.7071
2.7489	0.3827
3.1416	0.0000

Bài giải: xem trang 285

Bài 195: Viết một hàm Average(n, ...), nhận vào n tham số nguyên (sau tham số thứ nhất là n), trả về trung bình cộng của n tham số nguyên đó.



```
x = 32.4   y = 24.7   z = 4.5   t = 11.8
average( 2, x, y )      : 28.550
average( 3, x, y, z )   : 20.533
average( 4, x, y, z, t ) : 18.350
```

Bài giải: xem trang 285

Bài 196: Viết các macro:

islower(c) cho kết quả là 1 nếu ký tự c là chữ thường, hoặc 0 nếu ký tự c là chữ hoa.
 toupper(c) cho kết quả là ký tự hoa tương ứng với c nếu c là chữ thường, hoặc cho kết quả là chính ký tự c nếu c là chữ hoa.
 percent(a, b) cho kết quả là phần trăm của a trên b.



```
'c' viet hoa la 'C'
123 la 0.996355% cua 12345
```

Bài giải: xem trang 286

Bài 197: Viết các macro:

bitOff(d, n) đặt bit thứ n của d về 0.
 bitOn(d, n) bật bit thứ n của d lên 1.
 bitFlip(d, n) đảo bit thứ n của d.
 isBit(d, n) trả về 1 nếu bit thứ n của d đang là 1 hoặc trả về 0 nếu bit thứ n này đang là 0.

Với d là số nguyên.



Tạo rồi dùng macro isBit(d, n) để tạo hàm tạo in các bit của một số nguyên.



```
So goc   : 12345 = 00000000 00000000 00110000 00111001
Bat bit 8: 12601 = 00000000 00000000 00110001 00111001
Xoa bit 5: 12313 = 00000000 00000000 00110000 00011001
Dao bit 4: 12329 = 00000000 00000000 00110000 00101001
```

Bài giải: xem trang 286

Bài 198: Viết macro hbyte(n) và lowbyte(n) trả về byte cao và byte thấp của số integer n.



```
Platform 32-bits
INT_MAX / 10:
214748364 = 00001100 11001100 11001100 11001100
High Byte: 3276 = 00000000 00000000 00001100 11001100
Low Byte : 52428 = 00000000 00000000 11001100 11001100
```

Bài giải: xem trang 288

Bài 199: Tập tin văn bản EMP.TXT lưu thông tin nhân viên thành các dòng với định dạng sau:

ID:firstname lastname:birthday:salary

Các trường cách nhau bởi dấu ":" ; firstname và lastname cách nhau bởi dấu space; cuối dòng có ký tự newline.

Viết chương trình đọc tập tin EMP.TXT, hiển thị thông tin các nhân viên với firstname sắp xếp theo thứ tự alphabet, nếu firstname giống nhau, sắp xếp theo lastname. Thao tác sắp xếp phải sử dụng hàm chuẩn qsort() của stdlib.h.



```
Noi dung...
4424:Tom Jones:5/12/66:54335
2638:Jackie Lee:9/23/44:42500
1683:Billie Black:9/23/44:336500
1654:Jackie Chan:7/22/54:65000
5346:Mary Adams:11/4/63:28765
Sap xep...
1683:Billie Black:9/23/44:336500
1654:Jackie Chan:7/22/54:65000
2638:Jackie Lee:9/23/44:42500
5346:Mary Adams:11/4/63:28765
4424:Tom Jones:5/12/66:54335
```

Bài giải: xem trang 288

Bài 200: Tập tin văn bản PERSON.TXT lưu thông tin cá nhân thành các dòng với định dạng sau:

```
ID:firstname lastname:(area_code) phone_number
```

Các trường cách nhau bởi dấu ":", các cặp firstname và lastname, area_code (bao quanh bởi cặp ngoặc đơn) và phone_number, đều cách nhau bởi dấu space, cuối dòng có ký tự newline.

Viết chương trình đọc tập tin PERSON.TXT, nhập vào firstname của một người (giả sử các firstname đều khác nhau), tìm và hiển thị mã vùng của người đó. Thao tác tìm kiếm phải sử dụng hàm chuẩn bsearch() của stdlib.h.



```
Noi dung...
4424:Mike Harrington:(510) 5481278
2638:Christian Dobbins:(408) 5382358
5346:Jody Savage:(311) 5481278

Nhap firstname: Christian
Ma vung cua Christian: 408
```

Bài giải: xem trang 291

CẤU TRÚC DỮ LIỆU

Các bài tập cấu trúc dữ liệu (data structure) được đưa vào tập sách này được xem như là các bài tập dùng rèn luyện: thao tác trên con trỏ, truyền tham số bằng con trỏ (hiệu quả tương đương với truyền tham số bằng tham chiếu trong C++), giải thuật đệ quy...

Vì lý do trên, phần này chỉ giới hạn trong phạm vi các bài tập với danh sách liên kết đơn và cây tìm kiếm nhị phân (BST).

Theo quan điểm riêng của chúng tôi, bài tập cấu trúc dữ liệu được thực hiện bằng C++ hoặc Java với sự hỗ trợ của OOP sẽ dễ dàng và thuận lợi hơn. Tuy nhiên việc viết lại các bài tập trong phần này bằng C++ hoặc Java không khó.

Phần lớn các bài tập trong phần danh sách liên kết đơn đều dùng thông tin lưu tại

mỗi node (trừ các liên kết) là một số nguyên.

```
struct NODE {
    int data;
    struct NODE* next;
};
typedef struct NODE* NODEPTR;
```

Thông tin lưu trữ phức tạp hơn chỉ ảnh hưởng đến việc truy xuất, không ảnh hưởng đến giải thuật. Ví dụ:

```
struct STUDENT {
    char name[30];
    int age;
};
typedef struct STUDENT* SLINK;

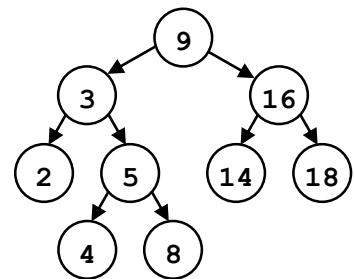
struct NODE {
    SLINK data;
    struct NODE* next;
};
typedef struct NODE* NODEPTR;
```

Nên tổ chức và quản lý thông tin lưu trữ như trên để thuận tiện cho việc hoán chuyển dữ liệu khi cần.

Phần lớn các bài tập trong phần BST đều dùng cấu trúc dữ liệu chung như sau:

```
struct NODE {
    int data;
    struct NODE *left, *right;
};
typedef struct NODE* NODEPTR;
```

Chúng tôi cũng dùng một cây có cấu trúc tương đối tổng quát để dễ kiểm tra kết quả xuất của các bài tập trong phần này, dữ liệu nhập: 9 3 16 14 2 5 4 8 18. Vì các bài giải dài và có nhiều tác vụ lặp đi lặp lại nên chỉ có một số bài giải đầy đủ, các bài giải còn lại chỉ tập trung giải quyết yêu cầu. Bài giải đầy đủ xin tham khảo trong source code của sách.



Bài 201: Xét các trường hợp chèn một node mới vào một danh sách liên kết đơn chứa các trị nguyên.



```
Nhap 0 de dung: 3 2 1 0 ↵
List goc: [1][2][3]
Chen 5 cuoi: [1][2][3][5]
Chen 4 sau node [3]: [1][2][3][4][5]
```

Bài giải: xem trang 296

Bài 202: Cho một danh sách liên kết đơn chứa các trị nguyên. Dùng giải thuật để quy, chèn một node chứa trị 0 sau mỗi node chứa trị chẵn trong danh sách liên kết.



```
Nhap 0 de dung: 1 2 4 3 5 6 0 ↵
List goc: [1][2][4][3][5][6]
Chen 0 sau tri chan: [1][2][0][4][0][3][5][6][0]
```

Bài giải: xem trang 300

Bài 203: Chèn một node mới vào một danh sách liên kết đơn, sao cho trị lưu trong các node của danh sách luôn theo thứ tự tăng. Không dùng đệ quy.



```
Nhap 0 de dung: 3 2 5 1 0 ↵
List goc: [1][2][3][5]
Nhap tri moi: 4 ↵
List moi: [1][2][3][4][5]
```

Bài giải: xem trang 301

Bài 204: Xóa một node trong một danh sách liên kết đơn chứa các trị nguyên. Xét các trường hợp phổ biến.



```
Nhap 0 de dung: 1 2 3 4 0 ↵
List goc: [1][2][3][4]
Nhap tri can xoa: 3 ↵
List moi: [1][2][4]
```

Bài giải: xem trang 302

Bài 205: Cho một danh sách liên kết đơn chứa các trị nguyên. Viết hàm thực hiện xóa lần lượt tất cả các node từ cuối danh sách ngược trở lên đầu.



```
Nhap 0 de dung: 1 2 3 0 ↵
List goc: [1][2][3]
Xoa node [3]
Xoa node [2]
Xoa node [1]
```

Bài giải: xem trang 304

Bài 206: Cho một danh sách liên kết đơn chứa các trị nguyên. Viết hàm thực hiện xóa tất cả các node có trị chỉ định nhập từ bàn phím.



```
Nhap 0 de dung: 1 2 1 1 3 1 4 1 1 0 ↵
List goc: [1][2][1][1][3][1][4][1][1]
Nhap tri can xoa: 1 ↵
List moi: [2][3][4]
```

Bài giải: xem trang 304

Bài 207: Cho một danh sách liên kết đơn chứa các trị nguyên, sửa các liên kết của danh sách này để có được hai danh sách danh sách liên kết mới: một danh sách chứa các trị lẻ của danh sách đã cho, một danh sách chứa trị chẵn của danh sách đã cho.



```
Nhap 0 de dung: 1 2 3 4 5 6 7 8 0 ↵
List goc : [1][2][3][4][5][6][7][8]
List chan: [8][6][4][2]
List le : [7][5][3][1]
```

Bài giải: xem trang 305

Bài 208: Cho một danh sách liên kết đơn chứa các trị nguyên. Đảo m phần tử cuối của danh sách liên kết lên đầu, m là số nguyên dương cho trước.



```
Nhap 0 de dung: 1 2 3 4 5 0 ↵
List goc: [1][2][3][4][5]
Nhap m: 2 ↵
List moi: [4][5][1][2][3]
```

Bài giải: xem trang 306

Bài 209: Cho một danh sách liên kết đơn chứa các trị nguyên (không chứa nhiều phần tử). In các phần tử của danh sách theo thứ tự ngược lại với thứ tự lưu trữ kể từ đầu danh sách. Tìm phần tử chứa trị lớn nhất trong danh sách bằng đệ quy.



```
Nhap 0 de dung: 1 3 5 2 4 0 ↵
List dao: [4][2][5][3][1]
Tri max: 5
```

Bài giải: xem trang 307

Bài 210: Đảo ngược các liên kết của một danh sách liên kết đơn chứa các trị nguyên.



```
Nhap 0 de dung: 1 2 3 4 5 0 ↵
List goc: [1][2][3][4][5]
List dao: [5][4][3][2][1]
```

Bài giải: xem trang 308

Bài 211: Cho danh sách liên kết đơn chứa các trị nguyên. Tách danh sách này thành các danh sách liên kết đơn con, mỗi danh sách liên kết đơn con chứa một “run” tăng. Dùng một danh sách liên kết đơn riêng quản lý các danh sách “run” nói trên.



```
Nhap 0 de dung: 1 2 3 2 3 4 5 4 6 0 ↵
List 'run':
r+--[4][6][n]
|-[2][3][4][5]
`-[1][2][3][n]
```

Bài giải: xem trang 309

Bài 212: Cho danh sách liên kết đơn quản lý các “run” như bài 211 (trang 61). Trộn các “run” do danh sách này quản lý thành một “run” tăng duy nhất.



```
Nhap 0 de dung: 1 2 3 2 3 4 5 4 6 0 ↵
List 'run':
r+--[4][6][n]
|-[2][3][4][5][n]
`-[1][2][3][n]
'run' tang duy nhat:
r+--[n]
|-[n]
`-[1][2][2][3][3][4][4][5][6][n]
```

Bài giải: xem trang 311

Bài 213: Tìm phần tử chứa trị xuất hiện nhiều lần nhất (tần suất cao nhất) trong một danh sách liên kết đơn khác rỗng chứa các trị nguyên, giả sử tần suất của các trị trong danh sách liên kết đều khác nhau.



```
Nhap 0 de dung: 1 2 3 3 4 3 2 2 4 2 0 ↵
List goc: [1][2][3][3][4][3][2][2][4][1]
Phan tu xuat hien nhieu nhat: [2](4)
```

Bài giải: xem trang 312

Bài 214: Cho một danh sách liên kết đơn chứa các trị nguyên. Thay đổi các liên kết của danh sách sao cho: các node chứa trị chẵn nằm đầu danh sách, các node chứa trị

lẻ nằm cuối danh sách, thứ tự xuất hiện các node giống với danh sách ban đầu.



```
Nhap 0 de dung: 3 8 4 1 5 7 6 2 0 ↵
List goc: [3][8][4][1][5][7][6][2]
List moi: [8][4][6][2][3][1][5][7]
```

Bài giải: xem trang 313

Bài 215: Cho một danh sách liên kết đơn không rỗng, viết chương trình thay đổi các liên kết của danh sách sao cho danh sách trở thành:

node[1] ⇒ node[3] ⇒ ... ⇒ node[2n+1]
 ⇒ node[2] ⇒ node[4] ⇒ ... ⇒ node[2n] ⇒ NULL



```
Nhap 0 de dung: 4 3 7 2 6 5 9 8 0 ↵
List goc: [4][3][7][2][6][5][9][8]
List moi: [4][7][6][9][3][2][5][8]
```

Bài giải: xem trang 314

Bài 216: Thực hiện giải thuật Selection Sort trên danh sách liên kết đơn chứa các trị nguyên, sắp xếp tăng các phần tử chứa trong danh sách.



```
Nhap 0 de dung: 1 3 5 7 2 4 6 8 0 ↵
List goc : [1][3][5][7][2][4][6][8]
List tang: [1][2][3][4][5][6][7][8]
```

Bài giải: xem trang 315

Bài 217: Cho một danh sách liên kết đơn chứa các trị nguyên. Đảo các node trong danh sách liên kết chứa trị k với node ngay sau nó. Trong đó, *hoán chuyển các node* thay vì hoán chuyển dữ liệu trong node.



```
Nhap 0 de dung: 1 2 3 1 2 3 1 0 ↵
List goc: [1][2][3][1][2][3][1]
Nhap k: 1 ↵
List moi: [2][1][3][2][1][3][1]
```

Bài giải: xem trang 315

Bài 218: Thực hiện giải thuật Selection Sort trên danh sách liên kết đơn chứa các trị nguyên, sắp xếp tăng các phần tử chứa trong danh sách. Trong đó, *hoán chuyển các node* thay vì hoán chuyển dữ liệu trong node.



```
Nhap 0 de dung: 1 3 5 7 2 4 6 8 0 ↵
List goc : [1][3][5][7][2][4][6][8]
List tang: [1][2][3][4][5][6][7][8]
```

Bài giải: xem trang 317

Bài 219: Thực hiện giải thuật Bubble Sort trên danh sách liên kết đơn chứa các trị nguyên, sắp xếp tăng các phần tử chứa trong danh sách.



```
Nhap 0 de dung: 1 3 5 7 2 4 6 8 0 ↵
List goc : [1][3][5][7][2][4][6][8]
List tang: [1][2][3][4][5][6][7][8]
```

Bài giải: xem trang 318

Bài 220: Thực hiện giải thuật Insertion Sort trên danh sách liên kết đơn chứa các trị nguyên, sắp xếp tăng các phần tử chứa trong danh sách.



```
Nhap 0 de dung: 1 3 5 7 2 4 6 8 0 ↵
List goc : [1][3][5][7][2][4][6][8]
List tang: [1][2][3][4][5][6][7][8]
```

Bài giải: xem trang 319

Bài 221: Xóa một node trong cây BST.



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
2 3 4 5 8 9 14 16 18
Nhap k: 5 ↵
2 3 4 8 9 14 16 18
```

Bài giải: xem trang 320

Bài 222: Thực hiện duyệt cây BST theo chiều sâu (deep-first traversal).



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
LNR: 2 3 4 5 8 9 14 16 18
NLR: 9 3 2 5 4 8 16 14 18
LRN: 2 4 8 5 3 14 18 16 9
```

Bài giải: xem trang 325

Bài 223: Thực hiện thao tác duyệt cây BST theo từng mức (breadth-first traversal).



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
BFS: 9 3 16 2 5 14 18 4 8
```

Bài giải: xem trang 327

Bài 224: Ý nghĩa của từng cách duyệt cây BST, ví dụ minh họa.



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
Cay goc : 9 3 2 5 4 8 16 14 18
Cay copy: 9 3 2 5 4 8 16 14 18
Xoa cay goc... 2 4 5 8 3 14 18 16 9
Cay goc rong
```

Bài giải: xem trang 329

Bài 225: Cho cây BST chứa các trị nguyên. Viết hàm thực hiện hai nhiệm vụ: xuất các node thuộc mức n và đếm các node thuộc mức n. Gốc có mức 0.



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
Nhap n: 2 ↵
Muc 2: 2 5 14 18
Tong: 39
```

Bài giải: xem trang 332

Bài 226: Cho cây BST chứa các trị nguyên. Tìm mức của node chứa trị x cho trước. Nếu cây không chứa trị x, thông báo không tìm thấy. Gốc có mức 0.



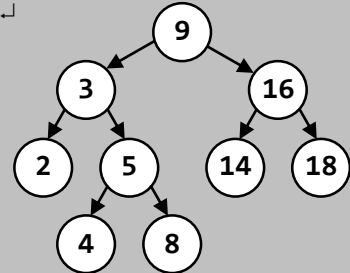
```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
Nhap x: 5 ↵
Muc 2
```

Bài giải: xem trang 333

Bài 227: Cho cây BST chứa các trị nguyên. Nhập các số x, y ($x < y$, có trong cây BST). Nếu một trong hai node chứa trị x hoặc y là node cha của node kia, trả về node cha. Nếu không, xác định cha chung gần nhất của hai node chứa x và y .



```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
Nhap a, b: 4 14 ↵
Node cha: 9
...
Nhap a, b: 16 14 ↵
Node cha: 16
...
Nhap a, b: 4 2 ↵
Node cha: 3
```



Bài giải: xem trang 333

Bài 228: Cho cây BST chứa các trị nguyên. Nhập các số x, y ($x < y$, có trong cây BST). Xác định đường đi từ node chứa x đến node chứa y .



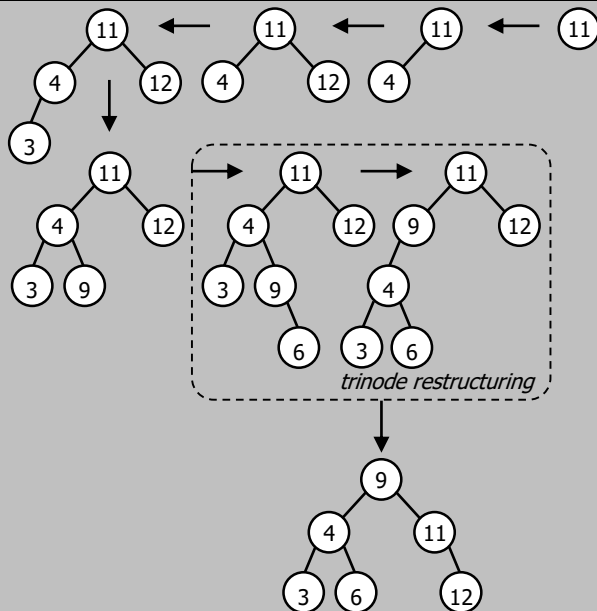
```
Nhap 0 de dung: 9 3 16 14 2 5 4 8 18 0 ↵
Nhap a, b: 18 5 ↵
5 3 9 16 18
...
Nhap a, b: 2 4 ↵
2 3 5 4
...
Nhap a, b: 9 14 ↵
9 16 14
```

Bài giải: xem trang 334

Bài 229: Thực hiện thao tác chèn một node mới vào cây nhị phân tìm kiếm cân bằng AVL. Tìm kiếm một node trong cây AVL trên.




```
(11, |)
-----
(11, \)
(4, |)
-----
(12, |)
(11, |)
(4, |)
-----
(12, |)
(11, \)
(4, \)
(3, |)
-----
(12, |)
(11, \)
(9, |)
(4, |)
(3, |)
-----
(12, |)
(11, /)
```



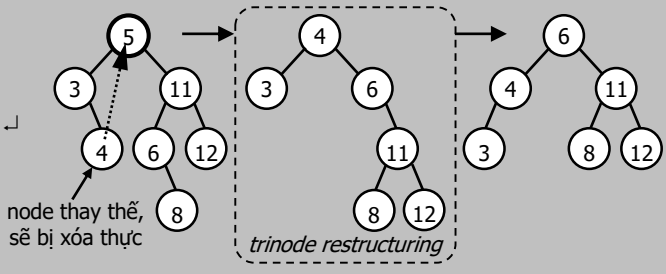

```
(9, |)
(6, |)
(4, |)
(3, |)
-----
Nhap khoa can tim: 12 ↵
[12, Heo]
```

Bài giải: xem trang 335

Bài 230: Thực hiện thao tác xóa một node có trong cây AVL.



Cay goc:
(12, |)
(11, \)
(8, |)
(6, /)
(5, /)
(4, |)
(3, /)
Khoa can xoa: 5 ↵
(12, |)
(11, |)
(8, |)
(6, |)
(4, \)
(3, |)



Bài giải: xem trang 340