CHƯƠNG 1: KIỂU DỮ LIỆU CÓ CẤU TRÚC – NHẬP XUẤT DỮ LIỆU TRÊN TẬP TIN

Ghi chú: Bài 1 đến 20 là bắt buộc.

1. Cho một lớp học gồm n học sinh (n≤50). Thông tin của một học sinh được định nghĩa theo kiểu dữ liệu của học sinh HOCSINH gồm:

- Mã số học sinh (MSHS): chuỗi có tối đa 5 ký tự.
- Họ tên (hoten): chuỗi có tối đa 30 ký tự.
- Ngày tháng năm sinh (ngaysinh): kiểu DATE.
- Địa chỉ (diachi): chuỗi có tối đa 50 ký tự.
- Giới tính (phai): chuỗi có tối đa 3 ký tự.
- Điểm trung bình (diemtb): số thực.

Hãy viết chương trình nhập và xuất danh sách học sinh sau đó đếm xem có bao nhiêu học sinh được lên lớp (Điều kiện được lên lớp là điểm trung bình ≥ 5.0).

Hướng dẫn:

- *Trước hết ta phải xây dựng hàm nhập và xuất cho 1 học sinh.*
- *Xây dựng hàm nhập và xuất ngày tháng năm (Kiểu dữ liệu DATE).*
- *Sau đó mới xây dựng hàm nhập và xuất cho danh sách học sinh.*

2. Cho một mảng các phân số (PHANSO) gồm n phần tử (n≤50). Hãy viết chương trình nhập và xuất danh sách các phân số sau đó tìm phân số có giá trị lớn nhất, tổng và tích các phân số và nghịch đảo giá trị các phân số trong mảng.

Hướng dẫn:

- *Trước hết ta phải xây dựng hàm nhập và xuất cho 1 phân số.*
- *Xây dựng hàm tính tổng, hiệu, tích, thương, rút gọn, so sánh và nghịch đảo cho 2 phân số.*
- *Sau đó mới xây dựng hàm nhập, xuất, tính tổng, tích cho mảng các phân số.*

3. Viết chương trình nhập vào 2 struct THOIGIAN (gồm giờ, phút, giây). Kiểm tra thời gian nhập vào có hợp lệ không, nếu hợp lệ thì xuất ra khoảng cách giữa 2 mốc thời gian trên.

4. Viết chương trình nhập vào 2 struct DATE (gồm ngày, tháng, năm), kiểm tra ngày nhập vào có hợp lệ hay không, nếu có thì xuất ra khoảng cách giữa 2 ngày.

5. Viết chương trình khai báo kiểu dữ liệu thể hiện một số phức. Sử dụng kiểu này để viết hàm tính tổng, hiệu, tích của hai số phức.

6. Viết chương trình khai báo kiểu dữ liệu để biểu diễn một phân số. Hãy viết hàm thực hiện những công việc sau:

a) Tính tổng, hiệu, tích, thương hai phân số.

b) Rút gọn phân số.

c) Qui đồng hai phân số.

d) So sánh hai phân số.

7. Viết chương trình khai báo kiểu dữ liệu để biểu diễn một hỗn số. Hãy viết hàm thực hiện những công việc sau :

a) Đổi hỗn số sang phân số

b) Tính tổng, tích hai hỗn số

8. Viết chương trình khai báo kiểu dữ liệu để biểu diễn một điểm trong hệ tọa độ 0xy . Hãy viết hàm thực hiện các công việc sau:

a) Tìm những điểm đối xứng của nó qua tung độ, hoành độ, toạ độ tâm.

b) Tính khoảng cách giữa hai điểm.

9. Viết chương trình tạo một mảng các số phức. Hãy viết hàm tính tổng, tích các số phức có trong mảng.

10. Viết chương trình tạo một mảng các phân số. Hãy viết hàm thực hiện các công việc sau :

a) Tính tổng tất cả các phân số (kết quả dưới dạng phân số tối giản)

b) Tìm phân số lớn nhất, phân số nhỏ nhất.

c) Sắp xếp mảng tăng dần.

11. Tổ chức dữ liệu để quản lí sinh viên bằng cấu trúc mẫu tin trong một mảng N phần tử, mỗi phần tử có cấu trúc như sau:

- Mã sinh viên.
- Tên.
- Năm sinh.
- Điểm toán, lý, hoá, điểm trung bình.

Viết chương trình thực hiện những công việc sau:

a) Nhập danh sách các sinh viên cho một lớp học.

b) Xuất danh sách sinh viên ra màn hình.
c) Tìm sinh viên có điểm trung bình cao nhất.
d) Sắp xếp danh sách lớp theo thứ tự tăng dần của điểm trung bình.
e) Sắp xếp danh sách lớp theo thứ tự giảm dần của điểm toán.
f) Tìm kiếm và in ra các sinh viên có điểm trung bình lớn hơn 5 và không có môn nào dưới 3.
g) Tìm sinh viên có tuổi lớn nhất.
h) Nhập vào tên của một sinh viên. Tìm và in ra các thông tin liên quan đến sinh viên đó (nếu có).

12. Tổ chức dữ liệu quản lí danh mục các bộ phim VIDEO, các thông tin liên quan đến bộ phim này như sau:

- Tên phim (tựa phim).
- Thể loại (3 loại : hình sự, tình cảm, hài).
- Tên đạo diễn.
- Tên điễn viên nam chính.
- Tên diễn viên nữ chính.
- Năm sản xuất.
- Hãng sản xuất

Viết chương trình thực hiện những công việc sau :

a) Nhập vào bộ phim mới cùng với các thông tin liên quan đến bộ phim này.
b) Nhập một thể loại: In ra danh sách các bộ phim thuộc thể loại này.
c) Nhập một tên nam diễn viên. In ra các bộ phim có diễn viên này đóng.
d) Nhập tên đạo diễn. In ra danh sách các bộ phim do đạo diễn này dàn dựng.

13. Một thư viện cần quản lý thông tin về các đầu sách. Mỗi đầu sách bao gồm các thông tin sau : MaSSach (mã số sách), TenSach (tên sách), TacGia (tác giả), SL (số lượng các cuốn sách của đầu sách). Viết chương trình thực hiện các chức năng sau:

a) Nhập vào một danh sách các đầu sách (tối đa là 100 đầu sách)
b) Nhập vào tên của quyển sách. In ra thông tin đầy đủ về các sách có tên đó, nếu không có thì tên của quyển sách đó thì báo là: Không Tìm Thấy

c) Tính tổng số sách có trong thư viện.

14. Viết chương trình tạo một mảng danh sách các máy tính của một cửa hàng, thông tin của một máy tính bao gồm :

- Loại máy
- Nơi sản xuất
- Thời gian bảo hành

a) Viết hàm nhập một dãy các loại máy tính có thông tin như trên.

b) Hãy viết hàm thống kê xem có bao nhiêu máy có thời gian bảo hành là 1 năm.

c) In ra danh sách các máy tính có xuất xứ từ Mỹ.

15. Để lắp ráp một máy vi tính hoàn chỉnh cần phải có tối thiểu 10 linh kiện loại A và có thể lắp bổ sung thêm vào khoảng tối đa 8 linh kiện loại B. Tại một cửa hàng vi tính cần quản lý bán hàng các loại linh kiện tại cửa hàng. Thông tin về một loại linh kiện gồm có: Tên linh kiện, quy cách, loại, đơn giá loại 1 (chất lượng tốt – số nguyên), đơn giá loại 2 (chất lượng thường – số nguyên). Viết chương trình thực hiện những công việc sau :

a) Nhập vào thông tin về các linh kiện có ở cửa hàng.

b) Xuất danh sách các linh kiện đã nhập theo thứ tự tăng dần của loại linh kiện và tên linh kiện.

c) Cho biết đã có đủ 10 linh kiện loại A cần thiết lắp ráp máy hay chưa?

16. Một cửa hàng cần quản lý các mặt hàng, thông tin một mặt hàng bao gồm:

- Mã hàng.
- Tên mặt hàng.
- Số lượng.
- Đơn giá.
- Số lượng tồn.
- Thời gian bảo hành (tính theo đơn vị tháng).

Hãy nhập vào một danh sách các mặt hàng.

a) Tìm mặt hàng có số lượng tồn nhiều nhất.

b) Tìm mặt hàng có số lượng tồn ít nhất.

c) Tìm mặt hàng có giá tiền cao nhất.

d) In ra những mặt hàng có thời gian bảo hành lớn hơn 12 tháng.

e) Sắp xếp các mặt hàng theo thứ tự tăng dần của số lượng tồn.

17. Viết chương trình tạo tập tin nhị phân chứa 10000 số nguyên bất kỳ ghi vào file SONGUYEN.INP, mỗi dòng 10 số. Sau đó viết chương trình đọc file SONGUYEN.INP, sắp xếp theo thứ tự tăng dần và lưu kết quả vào file SONGUYEN.OUT.

18. Viết chương trình tạo một file chứa 10000 số nguyên ngẫu nhiên đôi một khác nhau trong phạm vi từ 1 đến 32767 và đặt tên là "SONGUYEN.INP".

19. Viết chương trình tạo một file chứa các số nguyên có tên SONGUYEN.INP. Sau đó đọc file SONGUYEN.INP và ghi các số chẵn vào file SOCHAN.OUT và những số lẻ vào file SOLE.OUT.

20. Viết chương trình ghi vào tập tin SOCHAN.DAT các số nguyên chẵn từ 0 đến 100. Viết chương trình đọc tập tin SOCHAN.DAT và xuất ra màn hình, mỗi dòng 30 số.

CHƯƠNG 2: KỸ THUẬT TỔ CHỨC VÀ XỬ LÝ DỮ LIỆU TRÊN MẢNG 1 CHIỀU, 2 CHIỀU

Ghi chú: Bài 1 đến 12 là bắt buộc, bài 13 trở đi là khuyến khích.

Trình độ nhập môn

1. Tìm một số trong một mảng bằng lính canh.
2. Tìm một số trong một mảng đã có thứ tự (tìm nhị phân).
3. Viết các thủ tục thêm, xóa, sửa, tìm kiếm một phần tử trong một mảng.
4. Viết hàm chuyển một mảng hai chiều thành một mảng một chiều.
5. Viết hàm chuyển một mảng một chiều có MxN phần tử sang một mảng 2 chiều kích thước MxN.
6. Thực hiện ghép 2 mảng một chiều A, B để tạo mảng C theo nguyên tắc từng phần tử của mảng A xen kẻ từng phần tử của mảng B.
7. Thực hiện xóa bỏ khoảng trắng thừa và viết hoa đầu từ một chuỗi ký tự cho trước.
8. Cho ma trận A kích thước MxN (0<M,N<100) chứa các số thực nhỏ hơn 100000. Một điểm $X_{i,j}$ được gọi là điểm lồi nếu như nó lớn hơn cả 4 điểm trên, dưới, trái, phải của nó.

   Yêu cầu: Tìm X_{min} là điểm lồi có giá trị nhỏ nhất của mảng.

   Dữ liệu vào: Được nhập từ bàn phím có cấu trúc như sau:

   + Dòng đầu tiên là hai số nguyên dương M, N biểu diễn kích thước của ma trận A (M dòng, N cột).

   + M dòng tiếp theo, mỗi dòng là N số thực (mỗi số cách nhau ít nhất một khoảng trắng) lần lượt là N phần tử của từng dòng tương ứng của ma trận.

   Dữ liệu ra: Xuất ra màn hình một dòng duy nhất gồm 2 số nguyên I, J lần lượt là **chỉ số dòng và cột** của X_{min} đầu tiên từ trên xuống và từ trái qua phải. Nếu không có điểm lồi nào thì xuất ra là -1.

   Ví dụ:

Dữ liệu vào				**Dữ liệu ra**	
3	4			1	2
3	**9**	5	6		
4	6	**8**	7		
8	**11**	7	**10**		

9. Nhập vào một số dạng thập phân, chuyển sang dạng nhị phân, bát phân, thập lục phân.
10. Nhập vào d, m, y, kiểm tra (d,m,y) có lập thành một ngày tháng năm hay không, nếu có xuất ra ngày tiếp theo.

Kỹ thuật lập trình

11. Tính tổng, hiệu 2 số nguyên lớn.

<u>*Hướng dẫn*</u>*:*

- *Sử dụng kiểu dữ liệu chuỗi (mảng ký tự) cho từng số nguyên.*
- *Làm bài tính tổng trước, làm được mới tính hiệu, xử lý từng trường hợp 2 số có độ dài bằng nhau, độ dài khác nhau…*

12. Lập ma trận giá trị bãi mìn cho trò chơi Minesweeper.
13. Cho các số 1, 2, 3, 4, 5 và tổng S ban đầu bằng 0. Có 2 người chơi lần lượt chọn một trong các số đã cho theo nguyên tắc không được số mà người kia vừa chọn trước đó, mỗi lần ai chọn đều cộng dồn vào tổng S.

Ví dụ: Ban đầu người A chọn 2 vậy S=0+2=2.

Tiếp theo người B chỉ được chọn 1, 3, 4, 5 (không được chọn 2), ví dụ chọn 4, vậy S=2+4=6.

Tiếp theo A chỉ được chọn 1, 2, 3, 5 (được chọn lại 2 nhưng không được chọn 4), ví dụ chọn 5. Vậy S=6+5=11.

Tiếp theo B chỉ được chọn 1, 2, 3, 4 (không được chọn 5 vì A mới vừa chọn 5), ví dụ chọn 2. Vậy S=11+2=13….

Ai làm cho tổng S lớn hơn 35 là thua. Lập trình cho người chơi với máy sao cho khả năng thắng của máy là cao nhất.

<u>*Hướng dẫn*</u>*: Trò chơi được giải quyết bằng phương pháp Lập bảng phương án.*

14. Viết chương trình in ra các số nguyên từ 1 đến N^2 theo hình xoắn ốc với N được nhập vào từ bàn phím. Ví dụ, với N=4 ta có:

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

15. (**Ma trận kỳ ảo – Ma phương**) Viết chương trình nhập vào số tự nhiên N (N lẻ), sau đó điền các số từ 1 đến n^2 vào trong một bảng vuông sao cho tổng các hàng ngang, hàng dọc và 2 đường chéo đều bằng nhau.

Hướng dẫn: Một trong nhiều cách giải là như bên dưới

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

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

16. (**Bài toán dồn sỏi**). Cho k và n là 2 số nguyên dương, có 2^k viên sỏi, được phân bố trong n đống. Người ta cần san sẻ (theo quy tắc dưới đây) lượng sỏi từ các đống để dồn sỏi trở về một đống. Quy tắc san sỏi như sau: mỗi lần san áp dụng cho hai đống sỏi, giả sử rằng một đống có a viên sỏi và còn đống kia có b viên sỏi (không giảm tổng quát, giả thiết a, b) thì cho phép san (thay đổi số lượng từ hai đống) như sau: đống a viên (đống có số sỏi không thua đống còn lại) sẽ là a - b viên, còn đống b trở thành 2*b viên.

Hãy tìm cách san sỏi để cuối cùng còn 1 đống sỏi với 2^k viên.

17. (**Bài toán bốc sỏi từ hai đống sỏi**). Cho 2 đống sỏi, một đống có a viên sỏi, còn đống kia có b viên sỏi (a, b > 0). Hai đấu thủ lần lượt chơi trò bốc sỏi, mỗi lượt đi mỗi người được quyền bốc theo quy tắc:

- Ít nhất phải bốc được 1 viên
- Hoặc bốc một lượng sỏi bất kỳ từ một đống, hoặc bốc cùng một lượng sỏi như nhau ở hai đống (nếu đã bốc sỏi ở hai đống).

Đấu thủ đến lượt đi mà không còn sỏi để bốc thì coi như thua cuộc.

Tìm chiến lược đi để người trước thắng

18. Cho n là số tự nhiên. Cho trước một dãy gồm n ô liên tiếp nhau, mỗi ô hoặc được đánh dấu hoặc không được đánh dấu và chỉ ở một trong hai trạng thái nói trên mà thôi. Một trò chơi hai đấu thủ được mô tả như sau:

- Đầu tiên, toàn bộ n ô chưa bị đánh dấu,
- Hai đấu thủ luân phiên nhau đánh dấu vào các ô trong dãy theo quy tắc:
    - Hoặc chỉ đánh dấu vào một ô, hoặc đánh dấu vào hai ô liên tiếp nhau chưa bị đánh dấu.
    - Người đến lượt đi mà không còn ô chưa đánh dấu là bị thua.

Hướng dẫn:

- Trạng thái kết thúc: mọi ô đã bị đánh dấu,
- Tập trạng thái thua: Các miền (miền được hiểu theo nghĩa (1) gồm dãy liên tục các ô chưa đánh dấu, (2) ô bất kỳ, nếu có, phải bị đánh dấu trước) có thể ghép thành các cặp giống hệt nhau:
    - Số miền có 1 ô chưa đánh dấu là chẵn; số miền có 2 ô chưa đánh dấu là chẵn.
    - Chiến lược đi để người đầu tiên thắng:
        - Tại bước đi đầu tiên: nếu n lẻ đánh dấu vào ô có thứ tự (n div 2)+1" nếu n chẵn đánh dấu hai ô (n div 2) và (n div 2)+1.
        - (Sau đó đối thủ đi trước một bước),
        - Mỗi bước tiếp theo sẽ chọn một "bước đi đối xứng" với bước mà đối thủ vừa đi "Đối xứng" ở đây qua trục đối xứng của dãy.

Ví dụ, với n = 9 nếu đối thủ xoá đi ô 3 thì ta xóa ô 7, nếu đối thủ xóa đi 2 ô 1 và 2 thì ta xóa 2 ô 8 và 9 v.v. Với n=10, nếu đối thủ xóa ô 3 thì ta xoá ô 8, nếu đối thủ xóa hai ô 1 và 2 thì ta xoá 2 ô 9 và 10 v.v..

Sau mỗi bước đi của ta, trình trạng của dãy có tính đối xứng và vì vậy, hoặc đối thủ gặp trạng thái kết thúc, hoặc ta luôn còn nước đi.

Chú ý:

- Có thể đặt ra bài toán khó hơn: Từ một trạng thái bất kỳ (một số ô đã bị đánh dấu, còn các ô còn lại thì chưa), hãy tìm chiến lược đi để khả năng người đi trước thắng là cao nhất có thể được. Tương tự lý giải của trường hợp trên, chiến lược đi là đúng định hướng tới việc tạo ra các tình huống đối thủ vào trạng thái thua có thể được; nếu không đưa về “gần trạng thái thua hơn” và không để đối thủ buộc ta vào trạng thái thua. Một số ý tưởng liên quan đến vấn đề này:

- Sử dụng khái niệm ”miền” trước đây. Hai miền khác nhau được gọi là đi cặp với nhau nếu chúng cùng số lượng ô.
- Sau khi đã ghép được các cặp, còn một số miền chưa được ghép cặp: hãy khử số miền theo kiểu này song chú ý chớ nên tạo điều kiện cho đối thủ buộc ta rơi vào trạng thái thua.

19. Cho m, n là hai số tự nhiên. Cho trước một bảng hai chiều gồm m dòng, mỗi dòng có n cột các ô, mỗi ô được đánh dấu hoặc không được đánh dấu và chỉ ở một ttrong hai trạng thái nói trên mà thôi. Một trò chơi hai đấu thủ được mô tả như sau:

- Đầu tiên toàn bộ m x n ô của bảng chưa bị đánh dấu.
- Hai đấu thủ luân phiên nhau đánh dấu các ô trong dãy theo quy tắc sau:
  - Chỉ đánh dấu vào các ô chưa bị đánh dấu.
  - Đánh dấu từ 1 ô đến 4 ô chưa bị đánh dấu có kề cận đỉnh (hay cạnh) và nằm trọn trong một hình vuông có cạnh dài không quá 2.
- Người đến lượt đi mà không còn ô chưa bị đánh dấu để đi tiếp là bị thua.

Tìm chiến lược đi để người đi trước luôn thắng.

Hướng dẫn:

Hoàn toàn có thể sử dụng thuật toán đi đối xứng của bài trên để giải bài này. Có thể mở rộng bài toán như đã làm ở bài toán 3 song ở việc quản lý các miền khó khăn hơn rất nhiều.

20. Lập ma trận kỳ ảo theo cách khác bài 15 theo như hướng dẫn bên dưới.

Hướng dẫn:

Ví dụ: Với N=3 và N=5 ta có

2	7	6
9	5	1
4	3	8

Bắc

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

Nam

- Xuất phát từ ô bên phải của ô nằm giữa. Đi theo ***hướng đông bắc*** để điền các số 1, 2, ...
- Khi điền số, cần chú ý một số nguyên tắc sau:
  - Nếu vượt ra phía ngoài bên phải của bảng thì quay trở lại cột đầu tiên.
  - Nếu vượt ra phía ngoài bên trên của bảng thì quay trở lại dòng cuối cùng.
  - Nếu số đã điền k chia hết cho N thì số tiếp theo sẽ được viết trên cùng một hàng với k nhưng cách 1 ô về phía bên phải.

CHƯƠNG 3: TINH CHỈNH CHƯƠNG TRÌNH

Ghi chú: Bài 1 đến 12 là bắt buộc, bài 13 trở đi là khuyến khích.

Trình độ nhập môn

1. Nhập x và p, tính x^p.

2. Tính S(n) = 1 + (1+2) + (1+2+3)+…+(1+2+3+…+n) $(n<10^6)$

3. Tính $S(n)=1+\frac{1}{1+2}+\frac{1}{1+2+3}+\cdots\frac{1}{1+2+3+...+n}$

4. Tính S(n) = 1 + (1x2) + (1x2x3)+…+(1x2x3x…xn)

5. Tính $S(n)=1+\frac{1}{2!}+\frac{1}{3!}+\cdots\frac{1}{n!}$

6. Tính $e^x\approx 1+\frac{x}{1!}+\frac{x^2}{2!}+\cdots\frac{x^n}{n!}$ với x và n cho trước

7. Tính giá trị phần tử thứ n của dãy Fibonacci (không dùng mảng).

8. Nhập số thực A (0<A<4), tìm n nhỏ nhất thỏa

$$1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{n}>A$$

9. Nhập vào một mảng các số nguyên.

a/ Xếp lại mảng đó theo thứ tự giảm dần.

b/ Nhập vào một số nguyên từ bàn phím. Chèn số đó vào mảng sao cho mảng vẫn có thứ tự giảm dần. (không được xếp lại mảng)

10. (**Phân loại ký tự**) Cho một chuỗi ký tự, hãy phân loại mỗi ký tự theo 4 kiểu sau: kiểu chữ thường, kiểu chữ hoa, chữ số và kiểu "khác" (tức là các ký tự không thuộc ba loại trên).

Kỹ thuật lập trình

11. Viết chương trình thực hiện phép nhân 2 số nguyên lớn (từ 50-100 chữ số)

12. Tính S(n) = 1 + (1+2) + (1+2+3)+…+(1+2+3+…+n) $(n<10^{10})$

13. Viết chương trình sắp xếp tăng một mảng có n phần tử chỉ gồm các số 1,2 và 3 sao cho thực hiện ít phép hoán vị nhất.

14. Tính $S = A_0 + A_1x + A_2x^2 + \ldots + A_nx^n$

<u>Hướng dẫn</u>: Chương trình dưới đây dùng 2n phép tính nhân

```
S = A[0];
xi=1;
for (int i=1; i<=N;i++)
{
        xi = xi*x;
        S = S + A[i]*xi;
}
```

Liệu có cách viết nào tốt hơn (chạy nhanh hơn) không?

15. Biết giai thừa của n, kí hiệu: n! = 1 x 2 x 3 x … x n. Cho một số nguyên dương n. Hãy cho biết giai thừa của n có bao nhiêu chữ số ‘0’ ở bên phải.

Ví dụ: 5! = 1 x 2 x 3 x 4 x 5 = 120 có 1 chữ số ‘0’ ở bên phải.

16. Cho một chuỗi S chỉ gồm các ký tự < và > có chiều dài n (n<=1000). Yêu cầu: Hãy chèn n+1 số nguyên dương vào sao cho ta có bất đẳng thức đúng sao cho số nguyên lớn nhất Tmax trong n+1 số này là nhỏ nhất. Viết chương trình nhập vào chuỗi S và xuất ra Tmax như trên. Ví dụ: S = ><>< sẽ cho ra bất đẳng thức đúng như sau: 2>1<2>1<2. Vậy Tmax=2.

17. **(Dãy các số 1)** Cho một số nguyên n bất kỳ (0≤n≤10000), n không chia hết cho 2 và không chia hết cho 5. Hỏi có ít nhất bao nhiêu chữ số trong dãy các số 1 sao cho (dãy) số đó chia hết cho n.

18. Viết chương trình nhập vào một số nguyên n, xuất ra một số nguyên x >0 nhỏ nhất sao cho p= $\sum_{i=0}^{x-1} 1x10^i$, với p=nxb và b là một số nguyên dương.

Ví dụ: *n*=3 => x=3, n=7 => x=6, n=9901 => x=12

19. Tính $S(n) = (1+\frac{1}{1^2})(1+\frac{1}{2^2})\ldots(1+\frac{1}{n^2})$

BUỔI 4: PHƯƠNG PHÁP DUYỆT

Ghi chú: Bài 1 đến 18 là bắt buộc, bài 19 trở đi là khuyến khích.

Trình độ nhập môn

1. Tìm giá trị nhỏ nhất của các phần tử trong một mảng một chiều.
2. Tính tổng các phần tử trong mảng một chiều.
3. Tính n!
4. Viết chương trình tìm nhị phân (viết đệ quy)
5. Tính C_n^k
6. Tính giá trị phần tử thứ n của dãy Fibonacci (không dùng mảng)
7. Tính tổng giá trị các số nguyên có trong một chuỗi ký tự.

   Ví dụ: Chuỗi 2AS34ASDF342B có tổng là 2+34+342=378
8. Tìm số nguyên tố nhỏ nhất trong mảng.
9. Tìm UCLN của tất cả các phần tử trong mảng.
10. Tìm phần tử có tần số xuất hiện nhiều nhất trong một mảng.

Kỹ thuật lập trình

11. Liệt kê tất cả các dãy nhị phân có độ dài n.
12. Liệt kê tất cả tập con của tập n phần tử
13. Liệt kê tất cả hoán vị của tập n phần tử
14. Cài đặt bài mã đi tuần
15. Cài đặt bài Tháp Hà Nội.
16. Có N thành phố, được đánh số từ 0 đến N-1. Một người du lịch xuất phát từ một thành phố muốn đi thăm các thành phố khác, mỗi thành phố đúng một lần rồi quay về nơi xuất phát. Chi phí đi từ thành phố i đến thành phố j là A[i][j], ($0 \le i,j < N$). Hãy tìm một hành trình cho người du lịch để tổng chi phí theo hành trình này là ít nhất.
17. Có N chi tiết được đánh số từ 0 đến N-1 cần được gia công. Các chi tiết có thể hoàn thành trên một máy A hoặc trên một máy B. Các máy này có thể hoạt động độc lập và làm việc đồng thời. Biết rằng thời gian gia công chi tiết i trên máy A là A[i] và trên máy B là B[i]. Hãy tìm một phương án phân công cho các máy để thời gian hoàn thành cả N chi tiết là sớm nhất.

18. Một dãy dấu ngoặc hợp lệ là một dãy ký tự “(“ và “)” được định nghĩa như sau:

- Dãy rỗng là 1 dãy dấu ngoặc hợp lệ độ sâu là 0.
- Nếu A là dãy dấu ngoặc độ sâu k thì (A) là dãy dấu ngoặc hợp lệ độ sâu k + 1
- Nếu A và B là hai dãy dấu ngoặc hợp lệ với độ sâu lần lượt là p và q thì AB là dãy dấu ngoặc hợp lệ độ sâu là max (p,q)
- Độ dài của 1 dãy ngoặc là tổng số ký tự “(“ và “)”.

Hãy liệt kê tất cả các dãy dấu ngoặc hợp lệ có độ dài m và độ sâu n.

Dữ liệu vào	Dữ liệu ra
8 3	((()())) ((())()) ((()))() (()(())) ()((()))

Gợi ý: Một dãy dấu ngoặc hợp lệ là một dãy ký tự “(“ và “)” thoả cả hai điều kiện sau:
- Số dấu ngoặc mở và dấu ngoặc đóng bằng nhau.
- Duyệt từ trái sang phải, số dấu ngoặc mở luôn lớn hơn hay bằng số dấu ngoặc đóng ở mọi vị trí trong chuỗi.

19. Cài đặt bài toán 8 con hậu.

20. Cài đặt bài hình vuông Latinh (là hình vuông có dòng đầu và cột đầu gồm các số từ 1 đến n. Trên mỗi dòng và mỗi cột đều là một hoán vị của các phần tử từ 1 đến n.

CHƯƠNG 5: GIẢI THUẬT SINH

Ghi chú: Bài 1 đến 15 là bắt buộc, bài 16 trở đi là khuyến khích.

Trình độ nhập môn

1. Đổi tất cả các số thập phân từ 1 đến n sang hệ nhị phân.
2. Viết hàm tính tổng các phần tử là số Amstrong (là số có đặc điểm như sau: số có k ký số, tổng các lũy thừa bậc k của các ký số bằng chính số đó).

   Ví dụ: 153 là số có các ký số $1^3+5^3+3^3=153$
3. Viết chương trình tìm số lẻ nhỏ nhất nhưng lớn hơn mọi số chẵn trong mảng.
4. Viết hàm thực hiện các thao tác trên bit (bật, tắt, lấy giá trị bit thứ i của biến n).
5. Viết bitcount đếm số lượng bit 1 của một số nguyên dương n.
6. Cho hàm F(n) với n nguyên không âm được xác định như sau:

   F(0)=0, F(1)=1, F(2n)=F(n), F(2n+1) = F(n) + F(n+1).

   Viết chương trình tính F(n).
7. Viết chương trình xuất ra tất cả các số nguyên tố nhỏ hơn n theo thuật toán sàng Eratosthène.
8. Viết chương trình kiểm tra một chuỗi có đối xứng không?
9. Viết chương trình nhập vào ma trận A[M][N], hãy xuất ra màn hình các phần tử A[i][j] sao cho A[i][j] là phần tử có giá trị lớn nhất dòng i và nhỏ nhất cột j.

Kỹ thuật lập trình

10. Sinh tất cả các dãy nhị phân có độ dài n.
11. Sinh tất cả tập con của tập n phần tử.
12. Sinh tất cả hoán vị của tập n phần tử.
13. Viết chương trình sinh tất cả tổ hợp chập k của n phần tử cho trước.
14. Cho hàm F(n) với n nguyên không âm được xác định như sau:

    F(0)=0, F(1)=1, F(2n)=F(n), F(2n+1) = F(n) + F(n+1).

    Viết chương trình tính F(n) với điều kiện không dùng mảng độ dài N.
15. Hãy liệt kê tất cả các dãy nhị phân độ dài n mà trong đó cụm chữ số “01” xuất hiện đúng 2 lần.
16. Chú lùn Hugo đang bị lạc vào một mê cung hình chữ nhật gồm M dòng và N cột, M, N≤100. Các dòng (cột) đánh số từ 1 đến M (từ 1 đến N) từ trên xuống (từ trái

sang). Hugo đang đứng ở ô (X,Y). Từ một ô bất kỳ, trong mỗi bước di chuyển, Hugo có thể di chuyển đến 1 trong 8 ô chung quanh. Mỗi ô của mê cung ứng với một số trong phạm vi 0 đến 255 với ý nghĩa quy định những hướng Hugo có thể di chuyển từ ô đó. Quy định đó như sau:

Giả sử biểu diễn với 8 bit của số tại ô Hugo đang đứng (ghi chữ H) là $b_0b_1b_2b_3b_4b_5b_6b_7$, ta đánh số các ô chung quanh ô đó với các số 7..0, với 0≤I≤7, Hugo di chuyển theo hướng I nếu và chỉ nếu bit b_i=1.

Yêu cầu: Hãy chỉ cho Hugo một hành trình qua ít ô nhất để có thể thoát khỏi mê cung nếu có thể. Chú ý rằng Hugo có thể di chuyển ra một ô biên nhưng từ ô biên đó Hugo không đi ra ngoài mê cung được.

Dữ liệu: Vào từ file HUGO.INP trong đó dòng thứ nhất ghi 2 số M, N, tiếp theo là M dòng, dòng thứ I ghi N số tương ứng với các ô dòng thứ I của mê cung. Dòng cuối cùng ghi 2 số X,Y.

Kết quả: Ghi ra file HUGO.OUT như sau: dòng thứ nhất ghi số nguyên không âm C mà C=0 nếu Hugo không ra khỏi mê cung được, nếu C>0, đó chính là số ô trên hành trình Hugo đi ra khỏi mê cung. Nếu C>0, trong C dòng tiếp theo, mỗi dòng ghi chỉ số dòng và chỉ số cột của một ô lần lượt trên hành trình của Hugo bắt đầu từ ô (X,Y) và cuối cùng là ô trên biên mà từ ô đó Hugo có thể ra khỏi mê cung.

Ví dụ:

HUGO.INP	HUGO.OUT
5 6 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 3 4	3 3 4 4 4 5 4

17. Viết chương trình sinh tất cả chỉnh hợp chập k của n phần tử cho trước.

18. In ra theo thứ tự tăng dần tất cả các phân số tối giản 0<m/n<1 với mẫu số ≤10

19. Liệt kê tất cả các cách phân tích số nguyên dương n thành tổng các số nguyên dương, hai cách phân tích là hoán vị của nhau chỉ tính là một cách.

BUỔI 6: QUY HOẠCH ĐỘNG

Ghi chú: Bài 1 đến 12 là bắt buộc, bài 13 trở đi là khuyến khích.

Trình độ nhập môn

1. Đổi tất cả các số thập phân từ 1 đến n sang hệ nhị phân.

2. Tính $C_n^k = \dfrac{n!}{k!(n-k)!}$ với $0 \le k \le n \le 20$.

3. Viết chương trình xuất ra n phần tử đầu tiên của dãy Fibonacii

4. Viết hàm xóa các phần tử trùng nhau trong dãy, chỉ giữ lại một phần tử trong đó. Ví dụ: 1 2 3 2 1 2 4 -> 1 2 3 4

5. Viết chương trình tính tích của 2 ma trận

6. Viết chương trình đếm và liệt kê các mảng con tăng dần trong mảng một chiều các số nguyên.

Ví dụ: 6 5 3 2 3 4 2 7 các dãy con tăng dần là 2 3 4 và 2 7

7. Viết chương trình tìm mảng con tăng dần có tổng lớn nhất trong mảng một chiều.

8. Viết chương trình in ra tam giác Pascal

9. Viết chương trình đếm số lần xuất hiện của từng loại ký tự trong một chuỗi.

10. Viết chương trình đảo ngược các từ trong một chuỗi, mỗi từ được định nghĩa là cách nhau ít nhất một ký tự trắng. Ví dụ:

"TU DO HANH PHUC" -> "UT OD HNAH CUHP"

Kỹ thuật lập trình

11. (**Bài** toán **tìm dãy con đơn điệu dài nhất**). Cho N số nguyên dương, dãy $a_1, a_2, \ldots a_n$ gồm các số thực. Tìm trong dãy đó có một dãy con $a_{i1}, a_{i2}, \ldots a_{ik}$ của dãy trên thoã mãn điều kiện:

i) $a_{ij} \le a_{ij+1}$

ii) $i_j \le i_{j+1}$

iii) Không có dãy con nào thoả tính chất i) & ii), mà có nhiều phần tử hơn.

Hướng dẫn:

Đặt $A_0 = -\infty$ *và* $A_{n+1} = \infty$.

Gọi L[i] là độ dài dãy con tăng dài nhất của các phần tử lấy trong miền từ A_i *đến* A_{n+1} *và phần tử đầu của dãy tăng là* A_i. *Ta có công thức QHĐ để tính L[i] như sau:*

+ L[n+1] = 1

+ L[i] = max(L[j]) + 1 với mọi i<j<n+1 và $A_i \le A_j$

12. (**Xâu con chung dài nhất**) Cho hai xâu văn bản X và Y só độ dài nằm trong khoảng từ 50 đến 100. Sâu Z được gọi là "xâu con chung" của X và Y nếu Z nhận được từ X (hoặc Y) bằng cách loại bỏ một số kí tự nào đó. Ví dụ nếu X="AdksHKoiGAdksHKoiG" và Y="ADKSHKOIGADKSHKOIG" thì Z="AHK" là một con xâu chung của X và Y. Hãy tìm xâu con chung lớn nhất có thể được (trong ví dụ trên Z cực đại sẽ là "AHKGAHKG")

Hướng dẫn:

Gọi L[i][j] là độ dài xâu con chung dài nhất của xâu X[i] gồm i kí tự phần đầu của X và xâu Y[j] gồm j kí tự phần đầu của Y.

Ta có công thức QHĐ như sau:

+ L[0][j]=L[i][0]=0

+ L[i][j]=L[i-1][j-1] + 1 nếu X[i]=Y[j]

+ L[i][j]=max(L[i-1][j], L[i][j-1]) nếu X[i]<>Y[j]

13. Tính các phần tử của mảng C[n][k] = C_n^k = số tổ hợp chập k của n phần tử, với $0 \le k \le n \le 20$.

Biết $C_n^o = C_n^n = 1$ và $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$

Hướng dẫn:

Tính nghiệm của bài toán trong trường hợp riêng đơn giản nhất.

```
for (i=1;i<=n;i++)
{
        C[0][i] = 1;
        C[i][i] = 1;
}
```

Tìm các công thức đệ quy biểu diễn nghiệm tối ưu của bài toán lớn thông qua nghiệm tối ưu của các bài toán con.

```
for (i=2;i<=n,i++)
    for (j=1;i<i,j++)
        C[i][j] = C[i-1][j-1] + C[i-1][j];
```

C[n][k]	0	1	2	3	4	5
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

Có thể cải tiến: dùng 2 mảng một chiều thay cho 1 mảng hai chiều.

14. Cho n món hàng ($n \le 50$). Món thứ i có khối lượng là A[i] (số nguyên). Cần chọn những món hàng nào để bỏ vào một ba lô sao tổng khối lượng của các món hàng đã chọn là lớn nhất nhưng không vượt quá khối lượng W cho trước. ($W \le 100$). Mỗi món chỉ chọn 1 hoặc không chọn.

Input:

```
n     W
A[1]   A[2]   …    A[n]
```

Ví dụ:

```
4  10
5   2   4   3
```

OutPut:

Tổng khối lượng của các món hàng bỏ vào ba lô.

Khối lượng của các món hàng đã chọn.

Trong ví dụ trên:

Tổng khối lượng của các món hàng bỏ vào ba lô là 10

Khối lượng các món hàng được chọn: 5   2   3

<u>*Hướng dẫn*</u>*:*

Tổ chức dữ liệu:

Fx[k][v] là tổng khối lượng của các món hàng bỏ vào ba lô khi có k món hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v.

Với $k \in [1, n]$, $v \in [1, W]$.

Nói cách khác: Khi có k món để chọn, Fx[k][v] là khối lượng tối ưu khi khối lượng tối đa của ba lô là v.

Khối lượng tối ưu luôn nhỏ hơn hoặc bằng khối lượng tối đa: $Fx[k][v] \le v$

Ví dụ: Fx[4][10] = 8, nghĩa là trong trường hợp tối ưu, tổng khối lượng của các món hàng được chọn là 8, khi có 4 món đầu tiên để chọn (từ món thứ 1 đến món thứ 4) và khối lượng tối đa của ba lô là 10. Không nhất thiết cả 4 món đều được chọn.

Giải thuật tạo bảng:

** Trường hợp đơn giản chỉ có 1 món để chọn: Ta tính Fx[1][v] với mọi v:*

Nếu có thể chọn (nghĩa là khối lượng tối đa của ba lô >= khối lượng của các món hàng thứ 1), thì chọn: Fx[1][v] = A[1];

Ngược lại (v < A[1]), không thể chọn, nghĩa là Fx[1][v] = 0;

** Giả sử ta đã tính được Fx[k–1][v] đến dòng k–1, với mọi $v \in [1, W]$. Khi có thêm món thứ k để chọn, ta cần tính Fx[k][v] ở dòng k, với mọi $v \in [1,W]$*

Nếu có thể chọn món hàng thứ k (v >= A[k]), thì có 2 trường hợp:

– Trường hợp 1: Nếu chọn thêm món thứ k bỏ vào ba lô, thì

$$Fx[k][v] = Fx[k-1][u] + A[k];$$

Với u là khối lượng còn lại sau khi chọn món thứ k. u = v – A[k]

– Trường hợp 2: Ngược lại, không chọn món thứ k, thì

$$Fx[k][v] = Fx[k-1][v];$$

Trong 2 trường hợp trên ta chọn trường hợp nào có Fx[k][v] lớn hơn.

Ngược lại (v < A[k]), thì không thể chọn, nghĩa là Fx[k][v] = Fx[k–1][v];

Tóm lại: công thức đệ quy là:

```
if (v >= A[k])
        Fx[k][v] = Max(Fx[k-1][v - A[k]] + A[k] , Fx[k-1][v])
else
        Fx[k][v] := Fx[k-1][v];
```

Dưới đây là bảng Fx[k][v] tính được trong ví dụ trên:

[k][v]	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1	*0*	*0*	*0*	*0*	*5*	*5*	*5*	*5*	*5*	*5*
2	*0*	*2*	*2*	*2*	*5*	*5*	*7*	*7*	*7*	*7*
3	*0*	*2*	*2*	*4*	*5*	*6*	*7*	*7*	*9*	*9*
4	*0*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*

Giải thuật tra bảng để tìm các món hàng được chọn:

Chú ý: Nếu Fx[k][v] = Fx[k–1][v] thì món thứ k không được chọn.

Fx[n][W] là tổng khối lượng tối ưu của các món hàng bỏ vào ba lô.

Bước 1: Bắt đầu từ k = n, v = W.

Bước 2: Tìm trong cột v, ngược từ dưới lên, ta tìm dòng k sao cho

Fx[k][v] > Fx[k–1][v].

Đánh dấu món thứ k được chọn: Chọn[k] = true;

Bước 3: v = Fx[k][v] – A[k].

Nếu v > 0 thì thực hiện bước 2, ngược lại thực hiện bước 4

Bước 4: Dựa vào mảng, chọn để in ra các món hàng được chọn.

15. (**Bài toán chia kẹo**) Cho n gói kẹo (n ≤ 50). Gói thứ i có A[i] viên kẹo. Cần chia các gói kẹo này cho 2 em bé sao cho tổng số viên kẹo mỗi em nhận được chênh lệch ít nhất. Mỗi em nhận nguyên gói. Không mở gói kẹo ra để chia lại. Hãy liệt kê số kẹo trong các gói kẹo mỗi em nhận được.

Input:

n

A[1] A[2] … A[n]

Output: Số kẹo trong các gói kẹo mỗi em nhận được, và tổng số kẹo mỗi em nhận được.

<u>Hướng dẫn</u>:

Gọi S là tổng số viện kẹo S = A[1] + A[2] + … + A[n];

S2 là nửa tổng số kẹo: S2 = S/2; (chia nguyên)

Cho em bé thứ nhất chọn trước những gói kẹo sao cho tổng số viên kẹo mà em nhận được là lớn nhất nhưng không vượt quá số kẹo S2.

Gói kẹo nào em bé thứ nhất không chọn thì em bé thứ hai chọn.

Bài toán được đưa về bài ba lô 1.

16. (**Bài toán balô 2**) Cho n món hàng (n ≤ 50). Món thứ i có khối lượng là A[i] và giá trị C[i] (số nguyên). Cần chọn những món hàng nào để bỏ vào một ba lô sao tổng giá trị của các món hàng đã chọn là lớn nhất nhưng tổng khối lượng của chúng không vượt quá khối lượng W cho trước (W ≤ 100).

Mỗi món chỉ chọn 1 hoặc không chọn.

<u>Input:</u>

```
n    W
A[1]  C[1]
A[2]  C[2]
…
A[n]  C[n]
```

Ví dụ:

```
5 13
3 4
4 5
5 6
2 3
1 1
```

<u>Output</u>:

Tổng giá trị của các món hàng bỏ vào ba lô.

Khối lượng và giá trị của các món hàng đã chọn.

Trong ví dụ trên:

Tổng giá trị của các món hàng bỏ vào ba : 16

Các món được chọn: 1(3, 4) 2(4, 5) 3(5, 6) 5(1, 1)

<u>Hướng dẫn</u>:

Tương tự bài ba lô 1, nhưng Fx[k][v] là giá trị lớn nhất của ba lô khi có k món hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v.

Công thức đệ quy là:

if (v >= A[k])

*Fx[k][v] = Max(Fx[k-1][v-A[k]] + **C[k]**, Fx[k-1][v])*

else

Fx[k][v]:=Fx[k–1][v];

Chú ý: chỉ khác bài balô 1 ở chỗ dùng C[k] thay cho A[k]

Dưới đây là bảng Fx[k][v] tính được trong ví dụ trên:

[k][v]	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	4	4	4	4	4	4	4	4	4	4	4
2	0	0	4	5	5	5	9	9	9	9	9	9	9
3	0	0	4	5	6	6	9	10	11	11	11	15	15
4	0	3	4	5	7	8	9	10	12	13	14	15	15
5	1	3	4	5	7	8	9	10	12	13	14	15	16

17. (**Bài toán balô 3**) Cho n loại hàng (n ≤ 50). Mỗi món hàng thuộc loại thứ i có khối lượng là A[i] và giá trị C[i] (số nguyên). Số lượng các món hàng của mỗi loại không hạn chế. Cần chọn những món hàng của những loại hàng nào để bỏ vào một ba lô sao tổng giá trị của các món hàng đã chọn là lớn nhất nhưng tổng khối lượng của chúng không vượt quá khối lượng W cho trước (W ≤ 100). Mỗi loại hàng có thể hoặc không chọn món nào, hoặc chọn 1 món, hoặc chọn nhiều món.

Input:

```
n     W
A[1]  C[1]
A[2]  C[2]
…
A[n]  C[n]
```

Ví dụ:

```
5 13
3 4
4 5
5 6
2 3
1 1
```

OutPut:

Tổng giá trị của các món hàng bỏ vào ba lô.

Số lượng của các loại hàng đã chọn.

Trong ví dụ trên:

Tổng giá trị của các món hàng bỏ vào ba lô: 19

Các món được chọn:

Chọn 1 món hàng loại 1, mỗi món có khối lượng là 3 và giá trị là 4

Chọn 5 món hàng loại 4, mỗi món có khối lượng là 2 và giá trị là 3

<u>*Hướng dẫn*</u>:

Tổ chức dữ liệu:

Fx[k][v] là tổng giá trị của các món hàng bỏ vào ba lô khi có k loại hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v, với $k \in [1, n]$, $v \in [1, W]$.

X[k][v] là số lượng các món hàng loại k được chọn khi khối lượng tối đa của ba lô là v.

Giải thuật tạo bảng:

* *Trường hợp đơn giản chỉ có 1 món để chọn: Ta tính Fx[1][v] với mọi v:*

X[1][v] = v/A[1]; (chia nguyên)

*Fx[1][v] = X[1][v] * C[1]*

* *Giả sử ta đã tính được Fx[k–1][v] đến dòng k–1, với mọi* $v \in [1, W]$. *Khi có thêm loại thứ k để chọn, ta cần tính Fx[k][v] ở dòng k, với mọi* $v \in [1,W]$

*Nếu ta chọn xk món hàng loại k, thì khối lượng còn lại của ba lô dành cho các loại hàng từ loại 1 đến loại k – 1 là: u = v – xk * A[k]*

*Khi đó giá trị của ba lô là: Fx[k][v]= Fx[k–1][u] + xk * C[k]*

Với xk thay đổi từ 0 đến yk, ta chọn giá trị lớn nhất và lưu vào Fx[k][v]. Trong đó yk = v/A[k] (chia nguyên) là số lượng lớn nhất các món hàng loại k có thể được chọn bỏ vào ba lô, khi khối lượng tối đa của ba lô là v.

Tóm lại: công thức đệ quy là:

Fx[k][v] = Max(Fx[k-1][v – xk * A[k]] + xk * C[k])
Max xét với xk thay đổi từ 0 đến v/A[k] (chia nguyên), và v – xk * A[k] > 0

Dưới đây là bảng Fx[k][v] và X[k][v] tính được trong ví dụ trên. Bảng màu xám là X[k][v]:

[k][v]	1		2		3		4		5		6		7		8		9		10		11		12		13	
1	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3	12	3	12	3	16	4	16	4
2	0	0	0	0	4	0	4	0	5	1	8	0	9	1	9	1	12	0	13	1	14	2	16	0	17	1
3	0	0	0	0	4	0	4	0	5	0	8	0	9	0	10	1	12	0	13	0	14	0	16	0	17	0
4	0	0	0	0	4	0	4	0	7	1	8	0	10	2	11	1	13	3	14	2	16	4	17	3	19	5

5	0	0	1	1	4	0	5	1	7	0	8	0	10	0	11	0	13	0	14	0	16	0	17	0	19	0

18. (**Bài toán đổi tiền**): Cho n loại tờ giấy bạc. Tờ giấy bạc thứ i có mệnh giá A[i]. Số tờ mỗi loại không giới hạn. Cần chi trả cho khách hàng số tiền M đồng. Hãy cho biết mỗi loại tiền cần bao nhiêu tờ sao cho tổng số tờ là ít nhất. Nếu không đổi được, thì thông báo "KHONG DOI DUOC". N < 50; A[i] < 256; M < 10000

Input: n M
A[1] A[2] … A[n]

Ví du: 3 18
3 10 12

Output: Tổng số tờ phải trả.
Số tờ mỗi loại.

Hướng dẫn: (Cách 1, tương tự bài ba lô 3)

Gọi Fx[i][j] là số tờ ít nhất được dùng để trả số tiền j đồng khi có i loại tiền từ loại 1 đến loại i. Với i = 1 .. n; j = 1 .. M.

X[i][j] là số tờ giấy bạc loại thứ i được dùng chi trả số tiền j đồng.

** Trường hợp đơn giản chỉ có 1 loại tiền để chọn: Ta tính Fx[1][j] với mọi j*

$$Fx[1][j] = \begin{cases} j \text{ div } A[1] \text{ nếu } j \text{ mod } A[1] = 0 \\ \infty \text{ nếu } j \text{ mod } A[1] \neq 0 \text{ (không đổi được)} \end{cases}$$

** Giả sử ta đã tính được Fx[i–1][j] đến dòng i–1, với mọi j ∈ [1, M]. Khi có thêm loại tiền thứ i để chọn, ta cần tính Fx[i][j] ở dòng i, với mọi j∈[1, M]*

*Nếu ta chọn k tờ loại i, thì số tiền còn lại dành cho các loại tiền khác từ loại 1 đến loại i – 1 là: u = j – k * A[k]*

Khi đó tổng số tờ là: Fx[i][j]= Fx[i–1][u] + k

Với k thay đổi từ 0 đến kMax, ta chọn giá trị nhỏ nhất và lưu vào Fx[i][j]. Trong đó kMax = j div A[k] là số tờ nhiều nhất của loại tiền i để đổi số tiền j.

Tóm lại: công thức đệ quy là:

$$Fx[i,j] = Min(Fx[i-1, j - k * A[i]] + k)$$
*Min xét với k thay đổi từ 0 đến j div A[i], và j – k * A[i] > 0*

Hướng dẫn: (Cách 2)

Gọi Fx[i] là số tờ ít nhất được dùng để đổi số tiền i. Với i = 1 .. M.

Với quy ước Fx[i] = ∞ (hoặc 0) khi không đổi được.

X[i] là loại tiền cuối cùng được dùng đổi số tiền i. (chỉ lưu 1 loại tiền)

<u>Giải thuật tạo bảng</u>:

Xếp mệnh giá A[i] tăng dần.

Khởi gán Fx[i] = ∞, X[i] = 0 với mọi j = 1 .. M

Gán Fx[0] = 0

Với số tiền i chạy từ 1 đến M, ta tính Fx[i] và X[i], bằng cách:

Nếu chọn loại tiền j thì số tiền còn lại là i – A[j]

Fx[i] = Min(Fx[i – A[j]] + 1) nếu i >= A[j]

Min xét với loại tiền j chạy từ 1 đến n.

X[i] = j ứng với giá trị min của Fx[i]

Dưới đây là mảng Fx[i] và X[i] tính được trong ví dụ trên (dùng 3 loại tiền 3 đồng, 10 đồng, 12 đồng để đổi số tiền 18 đồng)

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Fx[i]	∞	∞	∞	1	∞	∞	2	∞	∞	3	1	∞	1	2	∞	2	3	∞	3
X[i]	0	0	0	1	0	0	1	0	0	1	2	0	3	2	0	3	2	0	3

19. (**Phân công kĩ sư** - *Đề thi tuyển sinh sau Đại học khoá 1997 Đại học Tổng hợp TP HCM*) Một cơ sở phần mềm có n phòng máy vi tính. Cơ sở này phải tuyển chọn m kĩ sư để bảo trì máy. Sau khi tham gia ý kiến của các chuyên gia và kinh nghiệm của các đơn vị khác, người ta hiểu rằng nếu phân công i kĩ sư chuyên bảo trì tại phòng máy j thì số máy hỏng hằng năm phải thanh lí là a[i,j]. Do hạn chế về thời gian và điều kiện đi lại chỉ có thể phân công mỗi kĩ sư bảo trì tại một phòng máy. Bảng ví dụ dưới đây với m = 5 (kĩ sư) và n = 3 (phòng máy).

Số kĩ sư	phòng máy 1	phòng máy 2	phòng máy 3
0	14	25	20
1	10	19	14
2	7	16	11
3	4	14	8
4	1	12	6
5	0	11	5

Yêu cầu: Tìm ra phương án phân công mỗi phòng máy phân bao nhiêu kĩ sư sao cho tổng số máy phải thanh lí hằng năm là ít nhất.

Dữ liệu: vào từ file văn bản KiSu.inp có 2 dòng:

- dòng đầu gồm 2 số nguyên dương m, n. (m, n < 50)
- m+1 dòng tiếp theo bảng a[i][j].

Kết quả: đưa ra file văn bản KiSu.out gồm 2 dòng:

- Dòng đầu chứa tổng số máy (ít nhất) phải thanh lí hằng năm.
- Dòng thư hai chứa n số nguyên dương là số kĩ sư được phân công bảo trì mỗi phòng máy.

Trong ví dụ trên, phân công 3 kĩ sư cho phòng máy 1, 1 kĩ sư cho phòng máy 2, và 1 kĩ sư cho phòng máy 3. Khi đó, hằng năm số máy ít nhất phải thanh lý là 37 máy.

Ví dụ:

KiSu.inp (đối với ví dụ trên)
5 3 14 25 20 10 19 14 7 16 11 4 14 8 1 12 6 0 11 5

KiSu.out
37 3 1 1

<u>Hướng dẫn</u>:

Gọi F[i][j] là số máy hư ít nhất hằng năm khi có i kĩ sư được phân công bảo trì j phòng máy đầu tiên.

19.1. (**Tam phân đa giác**) Cho một đa giác lồi n đỉnh. Hãy phân đa giác này thành n–2 tam giác bằng n – 3 đường chéo, sao cho tổng của độ dài của các đường chéo này là nhỏ nhất. Các đường chéo này không cắt nhau (chỉ có thể giao nhau ở đỉnh của đa giác).

Dữ liệu: vào từ file văn bản TAMPHAN.INP có n + 1 dòng:

- Dòng đầu chứa một số nguyên n là số đỉnh của đa giác (3 < n < 50).
- Mỗi dòng trong n dòng kế tiếp chứa hai số thực là hoành độ và tung độ của mỗi đỉnh của đa giác.

Kết quả: đưa ra file văn bản TAMPHAN.OUT, gồm dòng đầu chứa một số thực (có 4 chữ số thập phân) là tổng nhỏ nhất của độ dài của các đường chéo. Mỗi dòng trong n – 3 dòng tiếp theo chứa 2 số nguyên là chỉ số của hai đỉnh của mỗi đường chéo được chọn.

Ví dụ:

TAMPHAN.INP
6 2 1 2 4 6 6 10 6 10 3 7 0

TAMPHAN.OUT
17.4859 2 6 3 6 3 5

<u>Hướng dẫn</u>: Gọi Fx[i][j] là tổng độ dài ngắn nhất của các đường chéo khi tam phân đa giác có i đỉnh kể từ đỉnh thứ j.

20. (**Trạm bưu điện)** Trên một con đường thẳng, dài, số nhà của những nhà dọc theo một bên đường là số đo độ dài tính từ đầu con đường (số nguyên). Người ta chọn ra k nhà làm trạm bưu điện.

Hãy xác định số nhà của k nhà đó sao cho các nhà còn lại cách một trạm bưu điện nào đó là gần nhất hay tổng khoảng cách của các nhà còn lại đến một trạm bưu điện gần nhất nào đó là nhỏ nhất.

Dữ liệu: vào từ file văn bản **BuuDien.inp** gồm 2 dòng:

- Dòng đầu: chứa hai số nguyên n và k (n < 300; k < 30), với n là tổng số nhà trên con đường đó, k là số trạm bưu điện.
- Dòng thứ hai là các số nhà theo thứ tự tăng dần.

Kết quả: đưa ra file văn bản BuuDien.out gồm 2 dòng:

- Dòng đầu: là k nhà dùng làm trạm bưu điện.
- Dòng thứ hai là tổng khoảng cách của các nhà còn lại đến một trạm bưu điện gần nhất nào đó.

Ví dụ:

BuuDien.inp
10 5 1 2 3 6 7 9 11 22 44 50

BuuDien.out
2 7 22 44 50 9

21. Cho hai dãy số D1và D2 mỗi dãy có từ 50 đến 150 số nguyên dương. Dãy số D3 được gọi là "dãy con chung" củaD1 và D2 nếu D3 nhận được từ D1 (D2) bằng cách loại bỏ một số >=0 số nào đó. Tìm dãy chung con có số lượng số nhiều nhất.

Hướng dẫn: *Một dạng khác của bài toán xâu con chung*