

Lời nói đầu

Lập trình (còn gọi là *phương pháp quy hoạch*) là một kỹ thuật hữu dụng và quy tắc của bài toán tin học, đặc biệt là những bài toán tối ưu. Số lượng bài toán có giải bằng lập trình ngày càng tăng, ví dụ riêng kì thi Olympic quốc tế về Tin học IOI 2004 có tới 3 bài trong 6 bài thi có thể giải bằng lập trình. Những năm gần đây, trong hầu hết các thi chọn HSGQG đều có ít nhất 1 trong 3 bài có thể giải bằng phương pháp quy hoạch.

Nhóm tác giả chúng tôi biên tập tài liệu “*Bài tập quy hoạch*” này mong muốn giới thiệu lý thuyết và các bài tập đến những người học lập trình. Cuốn sách là tài liệu quý báu dành cho học sinh và sinh viên các ngành công nghệ thông tin và giáo viên môn Tin học các trường THPT.

Tài liệu gồm 4 chương:

Chương I: Cơ sở lý thuyết

Chương II: Một số bài tập cơ bản

Chương III: Bài tập chuyên sâu

Chương IV: Một số thuật toán

Chương I nêu rõ thuật ngữ, vị trí, ứng dụng của lập trình và cách nhận diện các bài tập có thể giải bằng phương pháp quy hoạch. Chương II phân tích và đưa ra chương trình giải các bài toán kinh điển như: Tìm dãy con không giảm dài nhất, Dãy con chung dài nhất, Tìm dãy con có tổng bằng S, Chương III, chương IV giới thiệu bài, cách giải, chương trình cài đặt những bài tập chuyên sâu.

Chúng tôi chân thành cảm ơn các bạn đồng nghiệp đã nhìn xét và góp ý cho bản thảo, trân trọng cảm ơn BGH trường THPT Chuyên Bắc Giang đã khích lệ, tạo điều kiện cho nhóm tác giả nghiên cứu tài liệu và làm ra mặt bản này.

Trong quá trình biên soạn, mặc dù chúng tôi đã cố gắng, song nội dung chuyên ngành càng có nhiều khía cạnh mới và sâu sắc nên chắc chắn tài liệu còn nhiều hạn chế. Chúng tôi rất mong các bạn xa gần góp ý, lần tái bản sau tài liệu sẽ hoàn thiện hơn.

Mời góp ý, xin gửi nhận xét:

Nhóm Tin học Trường THPT Chuyên Bắc Giang

Chương I: Cơ sở lý thuyết

I.1. Thuật toán chia nhỏ

I.1.1. Thuật toán chia nhỏ

Lưu trình cũng như chia nhỏ là các phương pháp giải bài toán bằng cách tách nhỏ bài toán thành các bài toán con nhỏ hơn.

Thuật toán *chia nhỏ* có thể coi là thuật toán thông dụng nhất trong Tin học. Khi giải bài toán P với kích thước ban đầu nào đó, nếu gặp trường hợp kích thước quá lớn, chúng ta thường nghĩ đến việc giải các bài toán tương tự như vậy với kích thước nhỏ hơn (gọi là các bài toán con của P). Thuật ngữ “chia nhỏ” thường chỉ đến hình thức “bước nhỏ dần” để giải quyết bài toán.

Chia nhỏ thường là “tách” một bài toán ban đầu thành các bài toán con nhỏ, sau đó giải các bài toán con này và tổng hợp để giải bài toán con ban đầu.

Nhưng giải bài toán con như thế nào? Chúng ta có thể thử chia nhỏ một cách ngẫu nhiên để “tách” bài toán con thành các bài toán con nhỏ hơn nữa, và tiếp tục hành động này cho đến khi gặp các bài toán con nhỏ mà có thể giải được. Các bài toán con được sinh ra sau mỗi lần “tách” cũng có thể cùng mức độ. Nhưng bài toán con sinh ra sau mỗi lần thì mức độ giảm dần (thấp hơn).

Thật ra quy luật là cách thức dùng và hiệu quả của thuật toán chia nhỏ. Quá trình quy luật để xây dựng các bài toán con vào ngăn xếp và sử dụng để giải các bài toán con theo thứ tự ngược lại bài toán ban đầu như trên như ngăn xếp cho đến khi giải được bài toán ban đầu đầy đủ ngăn xếp.

Ví dụ: Cho mảng $a[1..n]$ gồm các số nguyên. Tìm phần tử có giá trị lớn nhất trong mảng.

Xét chương trình thực hiện thuật toán tìm kiếm nhị phân (mảng chia
trình bày bên) giải bài toán trên. Trong chương trình chính có lời gọi
thực hiện tìm(1,n). Sau đây là thực hiện tìm(i,j):

Procedure tìm(i,j : integer); {Tìm x có trong mảng a[i .. j] hay không}

Var k : integer;

Begin

 K := (i+j) div 2;

 If a[k] = x then

 Begin

 Write('Có phần tử bằng ',x);

 Halt;

 End

 Else

 If (a[k] < x) and (k < j) then tìm(k+1,j)

 Else

 If (a[k] > x) and (k > i) then tìm(i,k-1)

End;

Do chỉ tìm kiếm trên mảng trong hai phần của “tách” ra, nên phức tạp
thuật toán là $O(\log_2 n)$, thời gian là tìm kiếm tuần tự.

I.1.2. Hình thức truy hồi

Giải bài toán bằng tiến hành theo nhu cầu. Kết quả của một bước
thông đưa vào kết quả của các bước thực hiện trước đó. Do vậy, cần xây
dựng hình thức thể hiện quan hệ với kết quả giữa các bước gọi là hình thức truy
hồi.

Ví dụ 1: Cho dãy số Fibonacci $\{F_1, F_2, \dots, F_n\}$, trong đó:

$F_1 = F_2 = 1$ và $F_n = F_{n-1} + F_{n-2}$ với $n > 2$ (**1**). Tìm số hạng thứ n.

Hệ thống tính số hạng thứ n của dãy Fibonacci là hệ thống **(1)** đã cho sẵn trong bài.

Ví dụ 2: Cho một dãy N số nguyên A_1, A_2, \dots, A_n . Hãy tìm cách xóa ít nhất số hạng nào trong dãy còn lại là dãy tăng hay nói cách khác hãy chọn một số hạng sao cho dãy B gồm các số hạng đó theo trình tự xuất hiện trong dãy A là dãy tăng.

Minh họa cho ví dụ 2:

Dãy A : 1, 4, 10, 9, 8, 17, 11, 7, 12, 6

Dãy B : 1, 4, 10, 11, 12

Phân tích lại bài toán ta thấy:

Khi dãy A có một phần tử thì dãy B có 1 phần tử.

Mở rộng bài toán với dãy A có 2 phần tử, có 3 phần tử, ... tìm độ dài dãy B . Ta có bài toán tổng quát: Gọi $F[i]$ là số phần tử của dãy con B khi dãy A có các phần tử từ 1 đến i .

Dễ dàng tính được:

$F[i] := 1$ nếu không tồn tại $j := 1..i-1$ sao cho $A[j] < A[i]$

$F[i] := \max\{F[j], j=1..i-1 \text{ thỏa mãn } a[j] < a[i]\} + 1$; **(2)**

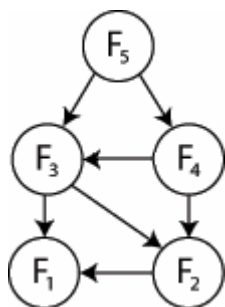
Như vậy, lại bài toán ví dụ 2 ta phải tiến hành qua n bước. Lại gọi b_i ($i=1, 2, \dots, n$) phụ thuộc vào lại gọi b_{i-1} bước trước đó. Hệ thống **(2)** chính là hệ thống truy hồi của bài toán.

I.1.3. Lập trình lại là gì?

Lập trình lại gọi là phương pháp chia nhỏ thành : lại gọi b_i của bài toán cũ để tính lại b_{i-1} của bài toán con.

“Chia nhỏ” sẽ phân chia bài toán ban đầu thành các bài toán con nhỏ (sự phân chia có cấu trúc đệ quy), gọi các bài toán con này thành bài toán quy, sau đó ghép lại b_i của chúng thành lại b_i của bài toán ban đầu.

Lập trình động (dynamic programming) cũng phân chia bài toán thành các bài toán con, nhưng các bài toán con *phân thu c nhau* (hay dùng từ gần nhau cũng được), mỗi bài toán con có thể tham chiếu tới cùng một số bài toán con mà nó đã đi (sự phân chia không có cấu trúc dạng cây xem hình 1).



Hình 1: Cấu trúc bài toán con cho dãy Fibonacci. Đây không phải là một **cấu trúc cây** mà là một **thực có hướng phi chu trình** mô tả quan hệ giữa các bài toán con gần nhau.

Nói rằng một bài toán có *các bài toán con trùng nhau* có nghĩa là mỗi bài toán con đó có thể được giải nhiều lần bởi các bài toán lớn hơn khác nhau. Ví dụ, trong **dãy Fibonacci**, $F_3 = F_1 + F_2$ and $F_4 = F_2 + F_3$ — khi tính mỗi số thì phải tính F_2 . Vì tính F_5 cần tính F_3 và F_4 , một cách tính F_5 một cách ngây thơ có thể sẽ phải tính F_2 hai lần hoặc nhiều hơn. Điều này áp dụng mỗi khi có một các bài toán con gần nhau: một cách tiếp cận ngây thơ có thể tốn thời gian tính toán lớn vì lặp lại việc tính toán cho các bài toán con mà nó đã giải.

Để tránh việc này, ta lưu trữ lại kết quả của các bài toán con đã giải. Do vậy, trong phần sau này ta cần giải lại chính bài toán đó, ta có thể lấy và sử dụng kết quả đã tính toán. Hình tiếp cận này được gọi là *lưu trữ* (trong **tiếng Anh** được gọi là **memoization**, không phải *memorization*, dù từ này cũng hợp nghĩa). Nếu ta chắc chắn rằng một lời giải nào đó không còn cần thiết nữa, ta có thể xóa nó đi để tiết kiệm không gian bộ nhớ. Trong một số trường hợp, ta còn có thể tính lại lời giải cho các bài toán con mà ta biết trước rằng sẽ cần.

Khi giải bài toán Fibonacci cấp i (với $i \geq 2$), ta sử dụng kết quả của 2 bài toán con cấp $i-1$ và $i-2$;

Lập trình động do nhà toán học người Mỹ là Richard Bellman (1920-1984) phát minh vào năm 1953.

Lập trình động thường dùng giải các bài toán tối ưu – bài toán yêu cầu tìm một giải pháp tốt nhất trong các giải pháp có thể tìm được.

Cơ sở của lập trình động trong bài toán tối ưu là nguyên lý tối ưu Bellman: “Dãy tối ưu các quyết định trong một quá trình quyết định nhiều giai đoạn có thuộc tính là dù trạng thái và các quyết định ban đầu bất kỳ thế nào, những quyết định còn lại phải tạo thành một cách giải quyết tối ưu không phụ thuộc vào trạng thái xảy ra từ những quyết định ban đầu”

I.1.4. Phương pháp quy hoạch động

Trong ngành khoa học máy tính, phương pháp quy hoạch động là một phương pháp giải một bài toán bằng cách chia các thuật toán thành các tính chất của các bài toán con *gặp nhau* và *cấu trúc con tối ưu*.

Tổng quát của phương pháp quy hoạch động có thể tóm lược như sau:

Chia 1 bài toán lớn thành các bài toán con nhỏ có kích thước nhỏ hơn cho đến khi nhận được các bài toán con mà lời giải có thể xây dựng dễ dàng.

Ta phải tìm ra mối quan hệ giữa nghiệm của bài toán với nghiệm của các bài toán con, mối quan hệ thể hiện 1 công thức truy hồi hay một hàm gọi là công thức quy hoạch động (hay hàm quy hoạch động).

Khi tính toán chúng ta xuất phát tính nghiệm các bài toán con ban đầu, sau đó kết hợp nghiệm của các bài toán con theo công thức quy hoạch động ta

được nghiệm bài toán con có kích thước lớn hơn. Quá trình tiếp tục như vậy cho đến khi ta được nghiệm của bài toán đã cho.

Ví dụ: Ý tưởng cơ bản của quy hoạch động thường là: tránh tính toán lại những thứ đã tính, mà lưu giữ kết quả đã tìm kiếm và dùng làm giá trị cho việc tìm kiếm những kết quả tiếp theo. Chúng ta sẽ làm rõ ý nghĩa của bài toán này bằng cách phân tích các kết quả của những trường hợp trước để hiểu rõ hơn. Kết quả cuối cùng chính là kết quả của bài toán cần giải.

Ví dụ: Tính số hạng thứ n của dãy Fibonacci bằng phương pháp quy hoạch động như sau:

Nghiệm của bài toán con chỉ cần tính 1 lần và lưu trữ trong mảng 1 chiều F . Khi tìm kết quả của các bài toán con liên tiếp, kết quả của các bài toán con chỉ cần lấy ra sử dụng.

$F1 := 1; F2 := 1;$

For $i := 3$ to n do $F[i] := F[i-2] + F[i-1];$

I.2. Các bước thể hiện giải bài toán bằng phương pháp quy hoạch động

I.2.1. Các bước cơ bản:

Bước 1: Phân tích tìm công thức quy hoạch động

- Tìm ra công thức nghiệm của bài toán liên thông qua việc giải các bài toán con.

- Xác định nghiệm của bài toán con cần giải (trong trường hợp cơ sở)

Bước 2: Thể hiện tính toán và thiết kế đệ quy

- Ưu tiên tính nghiệm của bài toán con cần giải (cơ sở)

(Khi biết giá trị ban đầu cho hàm QH)

- Đưa vào hàm QH đã xây dựng ta tính nghiệm của bài toán theo kích thước cần tìm cho đến khi nhận được nghiệm của bài toán đã cho.

Bước 3: Thiết kế đệ quy và cài đặt

- Thường dùng mảng để chứa hai chuỗi lưu trữ lại các giá trị của các bài toán con (giá trị của hàm QH)

- Ngoài ra, chúng ta phải dùng một số biến, thông thường cũng là mảng lưu trữ lại trạng thái của bài toán sau này chúng ta có thể truy xuất lại giá trị của bài toán.

I.2.2. Thuật cài đặt:

- Có thể dùng vòng lặp
- Hoặc dùng đệ quy
- Nghĩ về các bài toán con và tính toán lại và lưu trữ trong mảng khi cần lấy mảng đó ra và sử dụng

I.2.3 Khi nào dùng phương pháp quy hoạch động?

- Khi gặp bài toán có tính chất truy hồi
- Khi lời giải bài toán bước sau được xây dựng thông qua lời giải bài toán bước trước.
- Kích thước bài toán không quá lớn

* Ưu điểm:

- Lời giải ngắn gọn, sáng sủa, có tính logic cao, chương trình chạy nhanh.

* Nhược điểm:

- Không phải bài nào cũng giải bằng QH
- Việc tìm ra công thức QH của nhiều bài toán là khó khăn
- Không có công thức chung
- Bị hạn chế về bộ nhớ vì chúng ta phải lưu trữ lại các bài toán con.

Chương II: M t s bài t p c b n

Bài 1: Tìm dãy con không gi m nhi u ph n t nh t;

Cho dãy s nguyên có n ph n t a_1, a_2, \dots, a_n . M t dãy con không gi m c a dãy ã cho là dãy các ph n t còn l i c a dãy ó sau khi ta xóa b m t ho c m t s ph n t b t kì c a nó, các ph n t c a dãy con t o thành dãy không gi m. Ví d : dãy 1,4,10,11,12 là m t dãy con không gi m c a dãy 1, 4, 10, 9, 8, 17, 11, 7, 12, 6.

Yêu c u: Tìm dãy con không gi m c a dãy a g m nhi u ph n t nh t.

D li u: Vào t file v n b n DAYCON.INP g m:

Dòng u tiên ghi s N ($0 < N < 10^5$).

Các dòng ti p theo, m i dòng ghi 10 s nguyên c a dãy N s nguyên a_1, a_2, \dots, a_n . Dòng cu i cùng có th ít h n 10 s . Các s trên m t dòng cách nhau m t d u cách.

K t qu : Ghi ra file v n b n DAYCON.OUT g m:

S max là dài dãy con không gi m dài nh t tìm c.

Ch s xu t hi n c a các s h ng c a dãy con trong dãy ã cho.

Ví d :

DAYCON.INP	DAYCON.OUT
10	5
6 5 8 12 6 9 7 13 2 13	2 5 7 8 10

H ng d n:

G i $L(i)$ là dài dãy con t ng dài nh t, các ph n t l y trong mi n t a_1 n a_i và ph n t cu i cùng là a_i .

Ta có công th c QH tính $L(i)$ nh sau:

$$L(1) = 1$$

$$L(i) = \max(1, L(j)+1) \text{ v i m i ph n t } j: 0 < j < i \text{ và } a_j \leq a_i).$$

Bảng ghi nhận là mảng mảng chỉ u L lưu trữ các giá trị của hàm QH L(i). Ở đây chúng ta trình bày tính các giá trị của mảng L như sau:

```
for i := 1 to n do
begin
  L[i] := 1;
  for j:=1 to i - 1 do
    if (a[j]<=a[i]) and (L[i]<L[j]) then L[i]:=L[j]+1;
  end;
```

Như vậy chi phí không gian của bài toán là $O(n)$, chi phí thời gian là $O(n^2)$.

*** Chương trình cài đặt:**

```
program day_con_khong_giam_dai_nhat;
uses crt;
const fi='DAYCON.inp';
      fo='DAYCON.out';
      max=100000;
type  arrA=array[0..max]of longint;
      arrB=array[1..max,1..max]of longint;
var   f:text;
      a:arrA;
      n,kq,luu,dem:longint;
      d,truoc,k:arrA;
procedure  init;
var   i:longint;
begin
  assign(f,fi);
  reset(f);
```

```

    readln(f,n);
    for i:=1 to n do
        read(f,a[i]);
    close(f);
end;
procedure    process;
var    i,j:longint;
begin
    for i:=1 to n do
        truoc[i]:=i;
    d[1]:=1;
    for i:=1 to n do
        begin
            d[i]:=1;
            for j:=1 to i-1 do
                if a[i]>=a[j] then
                    if d[j]+1>d[i] then
                        begin
                            d[i]:=d[j]+1;
                            truoc[i]:=j;
                        end;
            end;
        end;
    kq:=0;
    for i:=1 to n do
        if d[i]>kq then begin kq:=d[i];luu:=i;end;
    dem:=0;

```

```

while truoc[luu]<>luu do
    begin
        inc(dem);
        k[dem]:=luu;
        luu:=truoc[luu];
    end;
inc(dem);
k[dem]:=luu;
end;
procedure inkq;
var i:longint;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,kq);
    for i:=dem downto 1 do
        write(f,k[i],' ');
    close(f);
end;
BEGIN
init;
process;
inkq;
END.

```

Bài 2: Dãy con chung dài nhất;

Cho hai dãy số nguyên $a = (a_1, a_2, \dots, a_M)$, $b = (b_1, b_2, \dots, b_N)$, với $M, N < 200$.

Hãy tìm một dãy con chung $c=(c_1, c_2, \dots, c_k)$ của a và b gồm nhiều số nguyên nhất. Dãy con chung c nhận được bằng cách xóa bớt một số số hạng của dãy a , c cũng nhận được bằng cách xóa bớt một số số hạng của dãy b , sau khi xóa hai dãy giữ nguyên thứ tự các phần tử còn lại.

D li u: Vào tệp file văn bản DAYCC.INP có cấu trúc như sau:

Dòng đầu tiên ghi 2 số nguyên dương M, N .

Dòng thứ 2 ghi các số a_1, a_2, \dots, a_M ;

Dòng thứ 3 ghi các số b_1, b_2, \dots, b_N ;

K t qu : Ghi ra file văn bản DAYCC.OUT có cấu trúc:

Dòng đầu ghi số K là số lượng các số hạng của dãy con chung c (nếu không có dãy con chung thì ghi số 0);

Trong k dòng tiếp theo, dòng thứ i ($i = 1, 2, \dots, k$) ghi 2 số x, y với ý nghĩa là: số hạng thứ i của dãy c là số hạng thứ x của dãy a và là số hạng thứ y của dãy b ;

Ví d :

DAYCC.INP	DAYCC.OUT
7 6	4
3 5 1 3 5 5 3	2 2
1 5 3 5 3 1	4 3
	6 4
	7 5

***H ãng d ãn:**

Gọi $L(i,j)$ là độ dài dãy con chung dài nhất của dãy a (i) gồm i phần tử phần đầu của dãy a ($a(i) = a[1..i]$) và dãy b (j) gồm j phần tử phần đầu của dãy b

(b(j)

=b [1..j]).

Ta có công thức quy hoạch động như sau:

$$L(0,j)=L(i,0)=0.$$

$$L(i,j) = L(i-1,j-1)+1 \text{ nếu } a[i] = b[j].$$

$$L(i,j) = \max(L(i-1,j), L(i,j-1)) \text{ nếu } a[i] \neq b[j].$$

***Chương trình cài đặt:**

Program Dayconchung;

uses crt;

const fi='daycc.inp';

fo='daycc.out';

maxmn=100;

var f:array[0..maxmn,0..maxmn] of integer;

g:text;

m,n:integer;

a,b:array[1..maxmn] of integer;

Procedure init;

var i:integer;

Begin

assign(g,fi); reset(g);

readln(g,m,n);

for i:=1 to m do read(g,a[i]);

for i:=1 to n do read(g,b[i]);

close(g);

End;

Function max(x,y:integer):integer;

Begin

```

    if x > y then max:=x else max:=y;
End;

Procedure Build;
var i,j:integer;
Begin
    for i:= 0 to m do f[i,0]:=0;
    for i:= 0 to n do f[0,i]:=0;
    for i:=1 to m do
        for j:=1 to n do
            if a[i] <> b[j] then f[i,j] := max(f[i-1,j],f[i,j-1])
            else    f[i,j] := f[i-1,j-1] + 1;
        end
    end
End;

Procedure result;
var i,j:integer;
Begin
    assign(g,fo); rewrite(g);
    writeln(g,f[m,n]);
    i := m; j := n;
    while (i<>0) and (j<>0) do
        Begin
            if a[i] = b[j] then
                Begin
                    writeln(g,i,' ',j);
                    dec(i);
                    dec(j);
                End
            else

```



```

Begin
    if f[i,j] = f[i-1,j] then dec(i)
    else if f[i,j] = f[i,j-1] then dec(j);
    end;
end;
close(g);
End;
BEGIN
    init;
    build;
    result;
END.

```

Bài 3: Dãy con có tổng bằng S

Cho dãy a_1, a_2, \dots, a_n . Tìm một dãy con của dãy có tổng bằng S.

D li u: vào tệp file văn bản "tongs.inp" có dạng:

- Dòng đầu tiên là 2 số N, S ($N \leq 200$);
- Dòng thứ hai là N số A_i ($i=1, 2, \dots, N$; $-32000 \leq A_i \leq 32000$).

K t qu : Ghi ra file văn bản "tongs.out" có dạng:

- Các phần tử thuộc dãy con tìm được

Ví d :

tongs.inp	Tongs.out
8 39	19 20
19 5 20 13 16 20 18 2	

*** H ướng d ẫn:**

t $L(i,t)=1$ n u có th t o ra t ng t t m t dãy con c a dãy g m các
ph n t a_1, a_2, \dots, a_i . Ng c l i thì $L(i,t)=0$. N u $L(n,S)=1$ thì áp án c a bài toán
trên là 'có'.

Ta có th tính $L(i,t)$ theo công th c: $L(i,t) = 1$ n u $L(i-1,t)=1$ ho c $L(i-1,t - a[i])=1$.

* Ch ng trình cài t:

```
Program day_con_tong_s;
uses crt;
const fi='tongs.inp';
      fo='tongs.out';
      max=6400;
      maxn=200;
type arrA=array[1..maxn]of longint;
      arrB=array[0..max]of longint;
var f:text;
    n:longint;
    a,truoc:arrA;
    s:longint;
    d:arrB;
procedure init;
var i:longint;
begin
  assign(f,fi);
  reset(f);
  readln(f,n,s);
  for i:=1 to n do
```

```

        read(f,a[i]);
    close(f);
end;
procedure process;
var i,j:longint;
begin
    for i:=1 to s do d[i]:=0;
    for i:=1 to n do
        for j:=s downto a[i] do
            begin
                if (d[j]=0) and (d[j-a[i]]=1) then
                    begin
                        d[j]:=1;
                        truoc[j]:=i;
                    end;
                if d[s]<>0 then exit;
            end;
        end;
    end;
end;
procedure inkq;
var i:longint ;
begin
    assign(f,fo);rewrite(f);
    if d[s]=0 then write(f,'0')
    else
        begin
            i:=truoc[s];

```

```

while s<>a[i] do
    begin
        write(f,a[i],' ');
        s:=s-a[i];
        i:=truoc[s];
    end;
write(f,a[i]);
end;
close(f);
end;
BEGIN
    init;
    process;
    inkq;
END.

```

Bài 4: Xếp vật vào ba lô (*mỗi vật chỉ có 1*)

Một chiếc ba lô có thể chứa tối đa W kg không quá W . Có N vật có khối lượng N , mỗi vật có khối lượng A_i và giá trị sử dụng C_i ($i=1, 2, \dots, N$). Hãy chọn các vật xếp vào ba lô sao cho tổng giá trị các vật trong ba lô là lớn nhất (nếu có thể xếp được).

Dữ liệu: vào tệp file văn bản BALO.INP có cấu trúc như sau:

Dòng đầu ghi 2 số nguyên dương N, W ($0 < N, W < 100$).

Dòng thứ $i+1$, với $i=1, 2, \dots, N$, ghi 2 số nguyên dương A_i và C_i tương ứng là khối lượng và giá trị sử dụng của vật thứ i ($0 < A_i, C_i < 256$).

Kết quả: Ghi ra file văn bản BALO.OUT có cấu trúc:

Dòng đầu ghi tổng giá trị sử dụng lớn nhất tìm được.

Dòng 2 ghi ch s các v t c x p vào ba lô.

Ví d :

BALO.INP	BALO.OUT
7 20	40
3 9	1 2 6 7
8 23	
23 4	
29 34	
12 4	
1 2	
8 6	

***H ãng d n**

G i $L(i,t)$ là t ãng giá tr l n nh t khi c ch n i v t t l n i cho vào balô v i t ãng kh i l ãng không v t quá t.

$L(n,m)$ s là áps c a bài toán (là giá tr l n nh t có c n u ch n n v t và t ãng kh i l ãng không v t m).

Công th c tính $L(i,t)$ nh sau:

$$L(i,0)=0; L(0,t)=0.$$

$$L(i,t)=L(i-1,t) \text{ n u } t < A_i.$$

$L(i,t)=\max(L(i-1,t), L(i-1,t-a_i)+b_i)$ n u $t \geq a_i$. Trong ó: $L(i-1,t)$ là giá tr có c n u không á v t i vào balô, $L(i-1,t-a_i)+b_i$ là giá tr có c n u ch n v t i.

Ta có th dùng m t m ãng 2 chi u l u b ãng ph ãng án, tuy nhiên d a trên nh n xét r ãng tính dòng i c a b ãng ph ãng án ch c n dòng $i-1$, ta ch c n dùng 2 m ãng m t chi u P và L có ch s t 0 n m l u 2 dòng ó. o n ch ãng trình con tính b ãng ph ãng án nh sau.

```

L[t] := 0; {v i m i t}
for l := 1 to n do begin
P:=L;
for t := 0 to m do
if t < L[t]:="P[t]" then > else L[t] := max(P[t],P[t-a[i]]);
end;

```

Nếu ý kiến s thay r ng o n trình trên ch vi t gi ng công th c QH ch ch a t i u. Ch ng h n ã có l nh gán $P:=L$, sau ó l i có gán $L[t]:=P[t]$ v i các giá tr $t < a[i]$ là không c n thi t. B n c có th t c i ti n ch ng trình t i u h n. Chi phí không gian c a cách cài t trên là $O(m)$ và chi phí th i gian là $O(n.m)$.

***Ch ng trình cài t:**

```

program xep_do_vat;
uses crt;
const fi='balo.inp';
      fo='balo.out';
      max=10000;
type arrA=array[1..max]of longint;
      arrB=array[0..max]of longint;
var f:text;
    n,w,kq,luu:longint;
    a,c :arrA;
    d,gt:arrB;
    truoc:arrA;
procedure init;
var i:longint;

```

```

begin
    assign(f,fi);
    reset(f);
    readln(f,n,w);
    for i:=1 to n do
        readln(f,a[i],c[i]);
    for i:=1 to w do
        begin
            d[i]:=0;
            truoc[i]:=0;
            gt[i]:=0;
        end;
    close(f);
    kq:=0;
end;

procedure    process;
var    i,j:longint;
begin
    d[0]:=1; gt[0]:=0;
    for i:=1 to n do
        begin
            for j:=w downto a[i] do
                if (d[j-a[i]]=1) then
                    if gt[j-a[i]]+c[i]>gt[j] then
                        begin
                            d[j]:=1;

```

```

        gt[j]:=gt[j-a[i]]+c[i];
        truoc[j]:=i;
        if gt[j]>kq then
            begin
                kq:=gt[j];
                luu:=j;
            end;
        end;
    end;
    write(i,' ',truoc[87]);
    readln;
end;

end;

procedure  inkq;
var  i :longint;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,kq);
    i:=truoc[luu];
    while luu>a[i] do
        begin
            write(f,i,' ');
            luu:=luu-a[i];
            i:=truoc[luu];
        end;
    write(f,i);

```



```
        close(f);  
end;  
BEGIN  
init;  
process;  
inkq;  
END.
```

Chương III: Bài tập chuyên đề

Bài 5: Bài tập phòng học;

Có n cuc ch p ánh s t l n n ng ký làm vi c t i m t phòng h i th o. Cuc ch p i c n c b t ut i th i i m s_i và k t thút t i th i i m f_i. H i có th b trí phòng h i th o ph c v c nhi u nh t bao nhiêu cuc ch p sao cho kho ng th i gian làm vi c c a hai cuc ch p c nh n ph c v ch có th giao nhau t i u nút.

D li u: vào t file v n b n ACTIVITY.INP

Dòng d u tiên ch a s nguyên d ng n ($n \leq 1000000$)

Dòng th i trong s n dòng ti p theo ch a hai s nguyên d ng s_i, f_i ($s_i, f_i \leq 32000$), $i=1,2, \dots, n$

K t qu : Ghi ra file v n b n ACTIVITY.OUT

Dòng u tiên ghi s k là s l ng cuc ch p c ch p nh n ph c v ;

M i dòng trong K dòng ti p theo ghi ch s c a m t trong k cuc ch p c ch p nh n

Ví d :

ACTIVITY.INP	ACTIVITY.OUT
5	3
1 3	1
2 4	4
1 6	5
3 5	
7 9	

*** H ạng d ạng:**

S ắp xếp các cu ố c h ợp t ổng d ạng theo th ứ i ếp m ột k ết thúc (b_i). Th ì cu ố c h ợp i s ẽ b ắt t ừ c ủa cu ố c h ợp j n ếu và ch ỉ n ếu $j < i$ và $b_j \leq a_i$. Yê u c ầ u b ắt t ừ c ủa cu ố c h ợp n ếu t ổng có th ể a ấ y v ề tìm d ãy các cu ố c h ợp dài nh ất tho ả m ẫu i ếp n ếu trên

*** Ch ương trình cài ặt:**

```
program   bo_tri_phong_hop;
uses crt;
const   fi='activity.inp';
        fo='activity.out';
        max=1000000;
type    arrA=array[1..max]of longint;
        arrB=array[1..max,1..max]of longint;
var     f:text;
        n,luu,kq,dem:longint;
        a,b,truoc,cs:arrA;
        d,k:arrA;
procedure   init;
var     i:longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            readln(f,a[i],b[i]);
```

```

        cs[i]:=i;
    end;
close(f);
end;
procedure    sap_xep;
var    i,j,tg:longint;ok:boolean;
begin
    for i:=1 to n-1 do
        begin
            ok:=true;
            for j:=n downto i+1 do
                if b[j]<b[j-1] then
                    begin
                        tg:=b[j];b[j]:=b[j-1];b[j-1]:=tg;
                        tg:=a[j];a[j]:=a[j-1];a[j-1]:=tg;
                        tg:=cs[j];cs[j]:=cs[j-1];cs[j-1]:=tg;
                        ok:=false;
                    end;
                if ok then exit;
            end;
        end;
    end;
end;
procedure    process;
var    i,j:longint;
begin
    d[1]:=1;
    for i:=1 to n do

```

```

truoc[i]:=i;
for i:=2 to n do
    begin
        d[i]:=1;
        for j:=1 to n-1 do
            if b[j]<=a[i] then
                if d[j]+1>d[i] then
                    begin
                        d[i]:=d[j]+1;
                        truoc[i]:=j;
                    end;
                end;
        end;
    kq:=-1;
    for i:=1 to n do
        if d[i]>kq then begin kq:=d[i];luu:=i;end;
    while luu<>truoc[luu] do
        begin
            inc(dem);
            k[dem]:=luu;
            luu:=truoc[luu];
        end;
    inc(dem);
    k[dem]:=luu;
end;
procedure    inkq;
var i:longint;

```

```

begin
    assign(f,fo);
    rewrite(f);
    writeln(f,kq);
    for i:=dem downto 1 do
        writeln(f,cs[k[i]]);
    close(f);
end;
BEGIN
init;
sap_xep;
process;
inkq;
END.

```

Bài 6: Cho thuê máy tính;

T i th i i m 0, ông ch cho thuê máy tính nh n c n t hàng thuê s d ng c a N khách hàng. Các khách hàng c ánh s t 1 n N. Khách hàng i c n s d ng máy t th i i m đi n th i i m ci (di và ci là các s nguyên và $0 < di < ci < 1000000000$), và s tr ti n s d ng máy là pi (pi nguyên, $0 < pi < 100000000$);

Yêu c u: Hãy xác nh xem ông ch c n nh n ph c v nh ng khách hàng nào sao cho kho ng th i gian s d ng máy tính c a hai khách hàng c nh n ph c v b t kì không giao nhau và t ng ti n thu c t ph c v là l n nh t.

D li u: vào t file v n b n RENTING.INP. Dòng u tiên ghi s N ($0 < N < 1000$). Dòng th i trong N dòng ti p theo ghi ba s di, ci, pi cách nhau b i d u tr ng, $i=1,2,...,N$.

K t qu : Ghi ra file v n b n RENTING.OUT. Dòng u tiên ghi hai s nguyên d ng theo th t là s l ng khách hàng nh n c ph c v và t ng ti n thu c. Dòng ti p theo ghi ch s c a khách hàng c ph c v .

Ví d :

RENTING.INP	RENTING.OUT	RENTING.INP	RENTING.OUT
3	2 180	4	2 1100
150 500 150	2 3	400 821 800	2 4
1 200 100		200 513 500	
400 800 80		100 325 200	
		600 900 600	

***H ng d n:**

T ng t nh bài toán 2, n u s p x p các n t hàng theo th i i m k t thúc, ta s a c bài toán 3 này v bài toán tìm dãy con có t ng l n nh t.

*** Ch ng trình cài t:**

```

program cho_thue_may_tinh;
uses crt;
const fi='renting.inp';
      fo='renting.out';
      max=1000;
type arrA=array[1..max]of int64;
      arrB=array[1..max,1..max]of longint;
var f:text;
    a,b,c:arrA;
    n,kq,luu,dem:longint;

```

```

        d,cs,truoc,k:arrA;
procedure    init;
var    i:longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            readln(f,a[i],b[i],c[i]);
            cs[i]:=i;
        end;
    close(f);
end;
procedure    sap_xep;
var    i,j,tg:longint;ok:boolean;
begin
    for i:=1 to n-1 do
        begin
            ok:=true;
            for j:=n downto i+1 do
                if b[j]<b[j-1] then
                    begin
                        tg:=b[j];b[j]:=b[j-1];b[j-1]:=tg;
                        tg:=a[j];a[j]:=a[j-1];a[j-1]:=tg;
                        tg:=cs[j];cs[j]:=cs[j-1];cs[j-1]:=tg;
                    end;
            end;
        end;
    end;
end;

```



```

        tg:=c[j];c[j]:=c[j-1];c[j-1]:=tg;
        ok:=false;
    end;
    if ok then exit;
end;
end;
procedure process;
var i,j:longint;
begin
    d[1]:=c[1];
    for i:=1 to n do
        truoc[i]:=i;
    for i:=2 to n do
        begin
            d[i]:=c[i];
            for j:=1 to n-1 do
                if b[j]<=a[i] then
                    if d[j]+c[i]>d[i] then
                        begin
                            d[i]:=d[j]+c[i];
                            truoc[i]:=j;
                        end;
            end;
        end;
    kq:=-1;
    for i:=1 to n do
        if d[i]>kq then begin kq:=d[i];luu:=i;end;

```

```

while luu<>truoc[luu] do
    begin
        inc(dem);
        k[dem]:=luu;
        luu:=truoc[luu];
    end;
inc(dem);
k[dem]:=luu;
end;
procedure inkq;
var i:longint;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,dem,' ',kq);
    for i:=dem downto 1 do
        write(f,cs[k[i]],' ');
    close(f);
end;
BEGIN  init; sap_xep;
        process;
        inkq;
END.

```

Bài 7: N i i m

Trên hai ng th ng song song L1 và L2 ng i ta ánh d u trên m i ng N i m. Các i m trên ng th ng L1 c ánh s t 1 n N t trái

sang phải, còn các vị trí trên hàng thứ hai có ánh sáng p_1, p_2, \dots, p_N cũng trải qua phải vị trí p_1, p_2, \dots, p_N là một hoán vị của $1, 2, \dots, N$ (Hình vẽ dưới đây cho một ví dụ khi $N=9$):

Ta gán các số gắn cho các vị trí là số hiệu của chúng. Cho phép nối hai vị trí trên hai hàng thứ hai có cùng số hiệu.

Yêu cầu tìm cách nối các hiệu để tìm một dãy liên tiếp các số nối không cắt nhau.

Dữ liệu: vào tệp văn bản WIRE.INP có cấu trúc như sau:

Dòng đầu tiên chứa số nguyên dương N ($N < 1000$);

Dòng tiếp theo chứa các số nguyên p_1, p_2, \dots, p_N cách nhau bởi dấu trống;

Kết quả: Ghi ra tệp văn bản WIRE.OUT có cấu trúc:

Dòng đầu tiên chứa số lượng các số nối tìm được;

Dòng tiếp theo chứa số hiệu của các đầu mút của các số nối được ghi theo thứ tự tăng dần.

Ví dụ:

WIRE.INP	WIRE.OUT
9	5
2 5 3 8 7 4 6 9 1	3 4 6 9

***Hàng đầu:**

Đây chính là bài toán 1: Tìm dãy con không giảm dài nhất

*** Chương trình cài đặt:**

```
program day_con;
```

```
uses crt;
```

```

const fi='wires.inp';
      fo='wires.out';
      max=1000;
type  arrA=array[0..max]of longint;
      arrB=array[1..max,1..max]of longint;
var   f:text;
      a:arrA;
      n,kq,luu,dem:longint;
      d,truoc,k:arrA;
procedure  init;
var   i:longint;
begin
      assign(f,fi);
      reset(f);
      readln(f,n);
      for i:=1 to n do
            read(f,a[i]);
      close(f);
end;
procedure  process;
var   i,j:longint;
begin
      for i:=1 to n do
            truoc[i]:=i;
      d[1]:=1;
      for i:=1 to n do

```

```

begin
    d[i]:=1;
    for j:=1 to i-1 do
        if a[i]>=a[j] then
            if d[j]+1>d[i] then
                begin
                    d[i]:=d[j]+1;
                    truoc[i]:=j;
                end;
            end;
    end;
    kq:=0;
    for i:=1 to n do
        if d[i]>kq then begin kq:=d[i];luu:=i;end;
    dem:=0;
    while truoc[luu]<>luu do
        begin
            inc(dem);
            k[dem]:=luu;
            luu:=truoc[luu];
        end;
    inc(dem);
    k[dem]:=luu;
end;
procedure inkq;
var i:longint;
begin

```

```

assign(f,fo);
rewrite(f);
writeln(f,kq);
for i:=dem downto 1 do
    write(f,a[k[i]],' ');
close(f);
end;
BEGIN
init;
process; inkq;
END.

```

Bài 8: Dãy con i chi u, i d u dài nh t;

Cho dãy a_1, a_2, \dots, a_n . Hãy tìm dãy con i chi u dài nh t c a dãy ó. Dãy con con i d u $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ph i tho mãn các i u ki n sau:

$a_{i_1} < a_{i_2} > a_{i_3} < \dots$ ho c $a_{i_1} > a_{i_2} < a_{i_3} > \dots$ các ch s ph i cách nhau ít nh t L : $i_2 - i_1 \geq L, i_3 - i_2 \geq L \dots$ chênh l ch gi a 2 ph n t liên ti p nh h n U : $|a_{i_1} - a_{i_2}| \leq U, |a_{i_2} - a_{i_3}| \leq U \dots$

Ví d : Cho dãy 10 ph n t : -1, 3, -4, 13, 6, 9, -2, 12, -3, 15 và $L=2$ và $U=3$

Có 1 dãy con i chi u dài nh t là: -1, -4, -2, -3

D li u: Vào t file v n b n DOICHIEU.INP có c u trúc nh sau:

Dòng u tiên ghi 3 s nguyên d ng n, L, U ($n < 10^5$; $1 < L < n-1$; $0 < U < 1000$);

Các dòng ti p theo ghi n s c a dãy a; ($|a_i| \leq 32000$);

K t qu : Ghi ra file v n b n DOICHIEU.OUT có c u trúc nh sau:

Dòng u là dài dãy con i chi u dài nh t tìm c;

Ch s các ph n t c a dãy con tìm c trong dãy ban u;

Ví dụ :

DOICHIEU.INP	DOICHIEU.OUT
10 2 3	4
-1 3 -4 13 6 9 -2 12 -3 15	1 3 7 9

*** Hàm tính :**

$G(i)$ là số phần tử của dãy con tăng dài nhất kết thúc tại a_i và phần tử cuối cùng lớn hơn phần tử trước. Tương tự, $P(i)$ là số phần tử của dãy con giảm dài nhất kết thúc tại a_i và phần tử cuối cùng nhỏ hơn phần tử trước. Ta dễ dàng suy ra:

$L(i) = \max(1, P(j)+1): j < i \text{ và } a_j < a_i.$

$P(i) = \max(1, L(j)+1): j < i \text{ và } a_i < a_j.$

*** Chương trình cài đặt:**

```
program day_con_doi_chieu_dai_nhat;
uses crt;
const fi='doichieu.inp';
      fo='doichieu.out';
      max=100000;
type arrA=array[1..max]of longint;
      arrB=array[1..max,1..max]of longint;
var f:text;
    n,l,u,luu,kq,cs,dem:longint;
    a:arrA;
    d1,d2,truoc1,truoc2,k:arrA;
procedure init;
var i:longint;
begin
```

```

assign(f,fi);
reset(f);
readln(f,n,l,u);
for i:=1 to n do
    begin
        read(f,a[i]);
        truoc1[i]:=i;
        truoc2[i]:=i;
    end;
close(f);
end;
procedure process;
var i,j:longint;ok:boolean;
begin
    d1[1]:=1;
    d2[1]:=1;
    for i:=2 to n do
        begin
            d1[i]:=1;
            for j:=1 to i-1 do
                if a[i]>=a[j] then
                    if abs(a[i]-a[j])<=U then
                        if d2[j]+1>d1[i] then
                            begin
                                d1[i]:=d2[j]+1;
                                truoc1[i]:=j;

```



```

                                end;
        d2[i]:=1;
        for j:=1 to i-1 do
            if a[i]<=a[j] then
                if abs(a[i]-a[j])<=U then
                    if d1[j]+1>d2[i] then
                        begin
                            d2[i]:=d1[j]+1;
                            truoc2[i]:=j;
                        end;
                    end;
                end;
            end;
        kq:=-1;
        for i:=1 to n do
            begin
                if d1[i]>kq then begin kq:=d1[i];cs:=1;luu:=i;end;
                if d2[i]>kq then begin kq:=d2[i];cs:=2;luu:=i;end;
            end;
        end;
    end;
    procedure    truy_xuat;
    var ok:boolean;
    begin
        ok:=false;
        dem:=0;
        repeat
            inc(dem);
            k[dem]:=luu;

```

```

    if cs=1 then
        begin
            if luu=truoc1[luu]then
                exit
            else
                begin
                    luu:=truoc1[luu];
                    cs:=2;
                end;
            end
        else
            begin
                if luu=truoc2[luu] then
                    exit
                else
                    begin
                        luu:=truoc2[luu];
                        cs:=1;
                    end;
                end;
            until ok;
        end;
    end;
procedure   inkq;
var   i:longint;
begin
    assign(f,fo);

```

```

rewrite(f);
writeln(f,kq);
for i:=dem downto 1 do
    write(f,k[i],' ');
close(f);
end;
BEGIN    init;  process;
        truy_xuat;  inkq;
END.

```

Bài 9: Số phép biến đổi ít nhất

Cho 2 chuỗi X,Y. Có 3 phép biến đổi với chuỗi X: chèn 1 ký tự, thay thế một ký tự hoặc xóa một ký tự. Hãy tìm số ít nhất các phép biến đổi biến chuỗi X thành chuỗi Y.

Ví dụ : Cho chuỗi X='kitten' và Y='sitting' thì số ít nhất là 3 phép biến đổi

D li u: Vào tệp file văn bản BIENDOI.INP có cấu trúc như sau:

Dòng 1 chứa chuỗi X

Dòng 2 chứa chuỗi Y

(Mỗi chuỗi có không quá 200 ký tự)

K t qu : Ghi ra tệp văn bản BIENDOI.OUT ghi 1 số là số phép biến đổi ít nhất tìm được;

Ví dụ :

BIENDOI.INP	BIENDOI.OUT
kitten	3
sitting	

*** H ướng dẫn:**

Giá trị $F(i,j)$ là số phép biến đổi ít nhất để biến chuỗi $X(i)$ giống với chuỗi $Y(j)$ (với $X(i) = X[1..i]$ và $Y(j) = Y[1..j]$). Do vậy $F(0,j)=j$ và $F(i,0)=i$.

Nếu $X[i]=Y[j]$ thì ta chỉ cần biến đổi chuỗi $X(i-1)$ thành chuỗi $Y(j-1)$. Do đó $F(i,j)=F(i-1,j-1)$.

Ngược lại, ta có 3 cách biến đổi:

- Xoá ký tự $X[i]$ và biến đổi chuỗi $X(i-1)$ thành $Y(j)$. Khi đó $F(i,j)=F(i-1,j)+1$.
- Thay thế $X[i]$ bởi $Y[j]$ và biến đổi $X(i-1)$ thành $Y(j-1)$. Khi đó $F(i,j)=F(i-1,j-1)+1$.
- Chèn $Y[j]$ vào $X(i)$ và biến đổi $X(i)$ thành $Y(j-1)$. Khi đó $F(i,j)=F(i,j-1)+1$.

Tổng kết lại, ta có công thức QH:

$$F(0,j)=j$$

$$F(i,0)=i$$

$$F(i,j) = F(i-1, j-1) \text{ nếu } X[i] = Y[j].$$

$$F(i,j) = \min(F(i-1,j), F(i,j-1), F(i-1,j-1)) + 1 \text{ nếu } X[i] \neq Y[j].$$

***Chương trình cài đặt:**

```
program bien_doi_xau;
uses crt;
const fi='biendoi.inp';
      fo='biendoi.out';
      max=100;
type arrA=array[1..max]of char;
      arrB=array[0..max,0..max]of longint;
var f:text;
     m,n:longint;
     d:arrB;
```

```

    a,b:arrA;
procedure    init;
begin
    assign(f,fi);
    reset(f);
    m:=0;
    while not eoln(f) do
    begin
        inc(m);read(f,a[m]);
        end;
        readln(f);
        n:=0;
        while not eof(f) do
        begin
            inc(n);read(f,b[n]);
            end;
            close(f);
        end;
function min3so(x,y,z:longint):longint;
var tam:longint;
begin
    if x<y then tam:=x
    else tam:=y;
    if z<tam then tam:=z;
    min3so:=tam;
end;

```

```

procedure    process;
var    i,j:longint;
begin
    for i:=1 to m do d[i,0]:=i;
    for j:=1 to n do d[0,j]:=j;
    for i:=1 to m do
        for j:=1 to n do
            begin
                if a[i]=b[j] then d[i,j]:=d[i-1,j-1]
                else
                    d[i,j]:=min3so(d[i-1,j-1],d[i,j-1],d[i-1,j])+1;
            end;
        end;
    end;
end;

procedure    inkq;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,d[m,n]);
    close(f);
end;

BEGIN
init;
process;
inkq;
END.

```

Bài 10: Xâu i x ng;

M t xâu g i là xâu i x ng (palindrom) n u xâu ó c t trái sang ph i hay t ph i sang trái u nh nhau. Cho m t xâu S, hãy tìm s kít ít nh t c n thêm vào S S tr thành xâu i x ng.

Ví d : Cho xâu S='abcda' thì s kít ít nh t c n chèn thêm là 2

D li u: Vào t file v n b n XDX.INP là 1 xâu S có không quá 500 kít)

K t qu : Ghi ra file v n b n XDX.OUT là s 1 ng kít ít nh t c n thêm;

Ví d :

XDX.INP	XDX.OUT
abcda	2

*H ng d n:

G i $L(i,j)$ là s kít ít nh t c n thêm vào xâu con $S[i..j]$ c a S xâu ó tr thành i x ng. áps c a bài toán s là $L(1,n)$ v i n là s kít c a S. Ta có công th c sau tính $L(i,j)$:

$$L(i,i)=0.$$

$$L(i,j)=L(i+1,j-1) \text{ n u } S[i]=S[j]$$

$$L(i,j)=\max(L(i+1,j), L(i,j-1)) \text{ n u } S[i] \neq S[j];$$

Ta có thu t toán n gi nh n nh sau:

G i P là xâu o c a S và T là xâu con chung dài nh t c a S và P. Khi ó các kít c a S không thu c T cũng là các kít c n thêm vào S tr thành i x ng. áps c a bài toán s là $n - k$, v i k là dài c a T. Ví d : S=edbabcd, xâu o c a S là P=dcbabde. Xâu con chung dài nh t c a S và P là T=dbabd. Nh v y c n thêm 2 kít là e và c vào S tr thành xâu i x ng;

*Ch ng trình cài t:

program xau_doi_xung;

```

uses crt;

const  fi='palin.inp';
       fo='palin.out';
       max=5000;

type  arrA=array[1..max]of char;
       arrB=array[1..max,1..max]of longint;

var   a:arrA;
       l:arrB;
       f:text;
       n:integer;

procedure  init;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do read(f,a[i]);
    close(f);
end;

function min2so(x,y:longint):longint;
begin
    if x>y then min2so:=y
    else min2so:=x;
end;

procedure  process;
var  i,j:longint;

```



```

begin
  for i:=1 to n do
    l[i,i]:=0;
  for i:=1 to n-1 do
    for j:=1 to n-i do
      begin
        if a[j]=a[j+i]then
          begin
            if i=1 then l[j,j+i]:=0
            else
              l[j,j+i]:=l[j+1,j+i-1];
            end
          else
            l[j,j+i]:=min2so(l[j+1,j+i],l[j,j+i-1])+1;
          end;
        end;
      end;
    end;
  end;
end;
procedure   inkq;
begin
  assign(f,fo);
  rewrite(f);
  writeln(f,l[1,n]);
  close(f);
end;
BEGIN
init;
process;

```

inkq;

END.

Bài 11: Bài toán chia kẹo

Có N gói kẹo, gói thứ i có A_i cái kẹo. Không cần bóc bỏ bất kỳ một gói kẹo nào, cần chia N gói kẹo thành hai phần sao cho chênh lệch số kẹo giữa hai gói là ít nhất.

D li u: Vào tệp văn bản "chiakeo.inp" có dạng:

- Dòng đầu tiên là số N ($N \leq 100$);
- Dòng thứ hai là N số A_i ($i=1, 2, \dots, N$; $A_i \leq 100$).

K t qu : Ghi ra tệp văn bản "chiakeo.out" có dạng:

- Dòng đầu là chênh lệch nhỏ nhất giữa hai phần có thể.
- Dòng thứ hai là các chênh lệch của các phần tử trong dãy a thu được phần

1;

- Dòng thứ ba là các chênh lệch của các phần tử trong dãy a thu được phần 2;

Ví d :

CHIAKEO.INP	CHIAKEO.OUT
8	1 56 57
19 5 20 13 16 20 18 2	3 5 6
	1 2 4 7 8

***H ớ ng d n:**

G i T là tổng số kẹo của gói. Chúng ta cần tìm số S lớn nhất tho mãn:

$$S \leq T/2.$$

Có một dãy con của dãy a có tổng bằng S. Khi đó sẽ có cách chia với chênh lệch 2 phần là $T - 2S$ là nhỏ nhất và dãy con có tổng bằng S trên gồm các phần tử là các gói kẹo thu được phần thứ nhất. Phần thứ hai là các gói kẹo còn lại.

*** Ch ớ ng trình cài t:**

program chia_keo;

```

uses crt;
const fi='chiakeo.inp';
      fo='chiakeo.out';
      max=100;
var   a:array[1..max]of integer;
      n:integer;
      t,k:integer;
      s:array[0..max,0..10000]of integer;
      f:text;
      t1,lech:integer;
      cx:array[1..max]of boolean;
procedure   init;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        read(f,a[i]);
    close(f);
    k:=0;
    for i:=1 to n do k:=k+a[i];
    t:=k div 2;
    for i:=1 to n do s[i,0]:=1;
    for i:=1 to t do s[0,i]:=0;
    for i:=1 to n do cx[i]:=true;

```

```

end;

procedure process;
var i,j:integer;
begin
    for i:=1 to n do
        for j:=1 to t do
            begin
                s[i,j]:=0;
                if s[i-1,j]=1 then s[i,j]:=1;
                if j>=a[i] then
                    if s[i-1,j-a[i]]=1 then s[i,j]:=1;
            end;
        end;
    end;

end;

procedure xu_li;
var i,j:integer;
begin
    for i:=t downto 1 do
        begin
            for j:=1 to n do
                if s[j,i]=1 then
                    begin
                        t1:=i;
                        lech:=k-2*i;
                        exit;
                    end;
            end;
        end;
    end;
end;

```

```

end;

procedure    truy_ket_qua;
var tam,i,t2:integer;
begin
tam:=t1;
t2:=n;
    repeat
        for i:=1 to t2 do
            if s[i,tam]=1 then
                begin
                    t2:=i;
                    tam:=tam-a[i];
                    cx[i]:=false;
                    break;
                end;
        until (tam=0);
end;

procedure    inkq;
var i:integer;
begin
assign(f,fo);
rewrite(f);
writeln(f,lech,' ',t1,' ',k-t1);
for i:=1 to n do if cx[i]=false then write(f,i,' ');
writeln(f);
for i:=1 to n do if cx[i]=true then write(f,i,' ');

```

```

close(f);
end;
BEGIN
init;
process;
xu_li;
truy_ket_qua;
inkq;
END.

```

Bài 12: Mua cá theo lý ng (Olympic Balkan 2000) ;

Ng i ánh cá Clement b t c n con cá, kh i l ng m i con là a_i , em bán ngoài ch . ch cá, ng i ta không mua cá theo t ng con mà mua theo m t l ng nào ó và m t con cá b t kì không c c t ra. Ch ng h n 3 kg, 5kg...

Ví d : có 3 con cá, kh i l ng l n l t là: 3, 2, 4. Mua l ng 6 kg s ph i l y con cá th 2 và và th 3. Mua l ng 3 kg thì l y con th nh t. Không th mua l ng 8 kg.

N u b n là ng i u tiên mua cá, có bao nhiêu l ng b n có th ch n?

D li u vào: c t file v n b n MARKET.INP có c u trúc:

- Dòng u tiên ghi s nguyên n ($n \leq 10^4$) ;
- Các dòng ti p theo ghi các giá tr a_i ($i=1,2,...,n$)

K t qu ra: Ghi ra file v n b n MARKET.OUT ghi s l là s l ng tìm c.

Ví d :

MARKET.INP	MARKET.OUT
3	7
3 2 4	

*** Hướng dẫn:**

Thức chất bài toán là tìm các số S mà có một dãy con của dãy a có tổng bằng S . Ta có thể dùng phương pháp ánh xạ của bài chia kẹo trên rổ để tìm các giá trị t mà $L[t]=1$.

*** Chương trình cài đặt:**

```
program so_ca;
uses crt;
const fi='market.inp';
      fo='market.out';
      maxn=10000;
      max=1000000;
type arrA=array[1..maxn]of longint;
      arrB=array[0..max]of longint;
var f:text;
    a:arrA;
    s,dem,n:longint;
    d:arrB;
procedure init;
var i:longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    s:=0;
    for i:=1 to n do
        begin
```

```

        read(f,a[i]);
        s:=s+a[i];
    end;
    close(f);
end;
procedure    process;
var    i,j:longint;
begin
    for i:=1 to s do
        d[i]:=0;
    d[0]:=1;
    for i:=1 to n do
        for j:=s downto a[i] do
            begin
                if (d[j]=0) and (d[j-a[i]]=1) then
                    begin
                        d[j]:=1;
                        inc(dem);
                    end;
            end;
        end;
    end;
end;
procedure    inkq;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,dem);
end;

```



```
close(f);
end;
```

Bài 13: i n d u h i vào bi u th c

Cho n s t nhiên a_1, a_2, \dots, a_n . Ban u các s c t liên ti p theo úng th t cách nhau b i d u '?': $a_1?a_2?...?a_n$. Cho tr c s nguyên S, có cách nào thay các d u '?' b ng d u + hay d u - c m t bi u th c s h c cho giá tr là S không?

D li u vào: c t file v n b n DAUHOI.INP có c u trúc nh sau:

Dòng u ghi 2 s nguyên d ng n và S ($n \leq 10^4, S \leq 10^5$);

Các dòng ti p theo ghi n giá tr a_1, a_2, \dots, a_n

K t qu ra: Ghi ra file v n b n DAUHOI.OUT có c u trúc nh sau:

Dòng u ghi 0 ho c 1 t ng ng là không có cách thay các d u h i ho c có t n t i cách thay d u h i b ng d u + hay - giá tr c a bi u th c b ng S;

N u dòng 1 ghi s 1 thì dòng ti p theo ghi 1 cách thay d u h i b i phép toán + ho c - tìm c (ch c n ghi các d u theo th t t trái sang ph i)

Ví d :

DAUHOI.INP	DAUHOI.OUT
5 20	1
4 5 6 7 8	-+++

*H ng d n:

t $L(i, t) = 1$ n u có th i n d u vào i s u tiên và cho k t qu b ng t.

Ta có công th c sau tính L:

$$L(1, a[1]) = 1.$$

$$L(i, t) = 1 \text{ n u } L(i-1, t+a[i]) = 1 \text{ ho c } L(i-1, t-a[i]) = 1.$$

N u $L(n, S) = 1$ thì câu tr l i c a bài toán là có. Khi cài t, có th dùng m t m ng 2 chi u (l u toàn b b ng ph ng án) ho c 2 m ng m t chi u (

l dòng i và dòng $i - 1$). Chú ý là ch s theo t c a các m ng ph i có c ph n âm (t c là t - T n T, v i T là t ng c a n s), vì trong bài này chúng ta dùng c d u - nên có th t o ra các t ng âm.

Bài này có m t bi n th là t d u sao cho k t qu là m t s chia h t cho k. Ta có thu t gi i t ng t bài toán trên b ng cách thay các phép c ng, tr b ng các hép c ng và tr theo mô un k và dùng m ng ánh d u v i các giá tr t 0 n k - 1 (là các s d có th có khi chia cho k). á p s c a bài toán là $L(n,0)$.

* Ch ng trình cài t:

```
program dau_hoi;
uses crt;
const fi='daudoio.inp';
      fo='dauhoi.out';
      max=10000;
      maxs=100000;
type arrA=array[1..max]of longint;
      arrB=array[0..maxs]of longint;
var f:text;
    n,s,tong:longint;
    a:arrA;
    d:arrB;
    ok:boolean;
    truoc:arrA;
    co:array[1..max]of boolean;
procedure init;
```

```

var
  i:longint;
begin
  assign(f,fi);
  reset(f);
  readln(f,n,s);
  tong:=0;
  for i:=1 to s do
    begin
      read(f,a[i]);
      tong:=tong+a[i];
      co[i]:=true;
    end;
  ok:=true;
  if tong<s then ok:=false;
  tong:=tong-s;
  if tong mod 2 <>0 then ok:=false
  else tong:=tong div 2;
  close(f);
end;
procedure process;
var i,j:longint;
begin
  for i:=1 to tong do
    d[i]:=0;
  d[0]:=1;

```

```

for i:=1 to n do
    for j:=tong downto a[i] do
        begin
            if (d[j]=0) and (d[j-a[i]]=1) then
                begin
                    d[j]:=1;
                    truoc[j]:=i;
                end;
            if d[tong]<>0 then exit;
        end;
    end;
end;

procedure inkq;
var luu,i:longint;
begin
    assign(f,fo);
    rewrite(f);
    if (d[tong]=0) or (truoc[tong]=1) then
        ok:=false
    else
        begin
            s:=tong;luu:=truoc[s];
            while s<>a[luu] do
                begin
                    co[luu]:=false;
                    s:=s-a[luu];
                    luu:=truoc[s];
                end;
            end;
        end;
end;

```

```

        end;
        co[luu]:=false;
    end;
if ok then
begin
writeln(f,'1');
for i:=2 to n do
    begin
        if co[i] then write(f,'+')
        else write(f,'-');
    end;
end
else write(f,'0');
close(f);
end;
BEGIN  init;
        if ok then  process;  inkq;
END.

```

Bài 14: Chia thành hai nhóm có tích l n nh t (ACM 10690)

Cho n số nguyên. Hãy chia chúng thành 2 nhóm sao cho tích của tất cả 2 nhóm là l n nh t.

Dòng vào: Đọc file văn bản EXPRESSION.INP có cấu trúc như sau:

- Dòng đầu ghi 1 số nguyên dương n ($n \leq 10^4$);
- Các dòng tiếp theo ghi n giá trị a_1, a_2, \dots, a_n

Kết quả: Ghi ra file văn bản EXPRESSION.OUT có cấu trúc như sau:

- Giá trị S là tích l n nh t của tất cả 2 nhóm tìm được;

Ví dụ :

EXPRESSION.INP	EXPRESSION.OUT
5 1 2 3 4 5	56

***Hạng độ n:**

Gọi T là tổng n số nguyên tố. Giả sử ta chia dãy thành 2 nhóm, gọi S là tổng của một nhóm, tổng nhóm còn lại là $T - S$ và tích của tổng 2 nhóm là $S*(T - S)$. Bằng phương pháp ánh xạ ta xác định S là tổng của một nhóm (như bài Market) và tìm S sao cho $S*(T - S)$ đạt max.

***Chương trình cài đặt:**

```
uses crt;
const fi='expression.inp';
      fo='expression.out';
      max=10000;
      maxs=100000;
type arrA=array[1..max]of longint;
      arrB=array[0..maxs]of longint;
var f:text;
     n,s,tong:longint;
     a:arrA;
     d:arrB;
     ok:boolean;
procedure init;
var i:longint;
```

```

begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            read(f,a[i]);
            tong:=tong+a[i];
        end;
    close(f);
end;
procedure    process;
var    i,j:longint;
begin
    for i:=1 to s do
        d[i]:=0;
    d[0]:=1;
    for i:=1 to n do
        for j:=s downto a[i] do
            begin
                if (d[j]=0) and (d[j-a[i]]=1) then
                    d[j]:=1;
                if d[s]<>0 then begin ok:=true;exit;end;
            end;
        end;
    end;
end;
procedure    xu_li;

```

```

begin
    s:=tong div 2 ;
    while not ok do process;
end;
procedure    inkq;
begin
    assign(f,fo);
    rewrite(f);
    if n<>1 then
        write(f,(s+1)*(tong-s-1))
    else writeln(f,'0');
    close(f);
end;
BEGIN
init;
if n<>1 then xu_li;
inkq;
END.

```

Bài 15: Bài toán mua bán hàng

Có m mặt hàng để mua hàng, anh ta có N đồng tiền d_1, d_2, \dots, d_N . Người bán hàng có M mặt hàng b_1, b_2, \dots, b_m . Anh ta muốn mua một mặt hàng với giá trị W . Hỏi cụ thể mua bán có thể diễn ra hay không?

Giới hạn: $M, N \leq 100$ và $d_i, b_j \leq 100$.

Dữ liệu vào: Đọc từ file văn bản MUABAN.INP có cấu trúc như sau:

- Dòng đầu ghi 3 số nguyên dương N, M, W
- Dòng tiếp theo ghi N số d_1, d_2, \dots, d_N ;

- Dòng thứ 3 ghi M s b_1, b_2, \dots, b_m ;

Kết quả ghi ra file văn bản MUABAN.OUT có cấu trúc như sau:

- Dòng đầu ghi số 0 hoặc 1 tùy thuộc là không thể đi ra cửa mua bán hoặc có thể đi ra cửa mua bán;
- Nếu dòng đầu là 1 thì dòng 2 ghi chỉ số các hàng mà người mua hàng đã sẵn sàng trao đổi, còn dòng 3 ghi chỉ số các hàng mà người bán hàng đã sẵn sàng trao đổi;

Ví dụ :

MUABAN.INP	MUABAN.OUT	MUABAN.INP	MUABAN.OUT
6 10 30	1	4 3 30	0
5 7 6 11 2 26	1 2 3 4 5	6 11 2 26	
1 2 3 5 6 7	1	10 20 30	
8 9 10			

* Chương trình cài đặt:

```

program mua_ban_hang;
uses crt;
const fi='muaban.inp';
      fo='muaban.out';
      max=100;
      maxs=10000000;
type arrA=array[1..max]of longint;
      arrB=array[0..maxs]of longint;
var f:text;
     a,b:arrA;
     d:arrB;
     n,m,w,tong,luu1,luu2,s,v:longint;
     ok1,ok2,ok:boolean;

```

```

        truoc1,truoc2:arrA;
procedure    init;
var i:longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n,m,w);
    tong:=0;
    for i:=1 to n do
        begin
            read(f,a[i]);
            tong:=tong+a[i];
        end;
    readln(f);
    for i:=1 to m do
        read(f,b[i]);
    close(f);
end;
procedure    process1;
var  i,j:longint;
begin
    for i:=1 to s do
        begin
            d[i]:=0;
            truoc1[i]:=0;
        end;

```

```

d[0]:=1;
for i:=1 to n do
    for j:=s downto a[i] do
        begin
            if (d[j]=0) and (d[j-a[i]]=1) then
                begin
                    d[j]:=1;
                    truoc1[j]:=i;
                end;
            if d[s]<>0 then begin ok1:=true;exit;end;
        end;
    end;
end;
procedure    process2;
var    i,j:longint;
begin
    d[0]:=1;
    for i:=1 to v do
        begin
            d[i]:=0;
            truoc2[i]:=0;
        end;
    for i:=1 to m do
        for j:=v downto b[i] do
            begin
                if (d[j]=0) and (d[j-b[i]]=1) then
                    begin

```

```

        d[j]:=1;
        truoc2[j]:=i;
    end;
    if d[v]<>0 then begin ok2:=true; exit;end;
end;
end;
procedure    xu_li;
begin
    ok:=false;
    for s:=w to tong do
        begin
            ok1:=false;
            process1;
            if ok1 then
                begin
                    ok2:=false;
                    v:=s-w;
                    if v=0 then ok2:=true
                    else
                        process2;
                end;
            if ok1 and ok2 then
                begin
                    luu1:=s;
                    luu2:=v;
                    ok:=true;

```

```

                break;
            end;
        end;
    end;
end;
procedure    inkq;
var i:longint;
begin
    assign(f,fo);
    rewrite(f);
    if ok then
    begin
        writeln(f,'1');
        i:=truoc1[luu1];
        while luu1<>a[i] do
            begin
                write(f,i,' ');
                luu1:=luu1-a[i];
                i:=truoc1[luu1];
            end;
        writeln(f,i);
        if luu2<>0 then
        begin
            i:=truoc2[luu2];
            while luu2<>b[i] do
                begin
                    write(f,i,' ');

```

```

        luu2:=luu2-b[i];
        i:=truoc2[luu2];
    end;
    write(f,i);
    end;
end
else writeln(f,'0');
close(f);
end;
BEGIN
init;
xu_li;
inkq;
END.

```

Bài 16: L ch thuê nhân công

Có m t d án kéo dài trong T tháng và ng i qu n lý c n ph i l p l ch s d ng công nhân trong d án, anh ta bi t s công nhân t i thi u c n trong m i tháng. M i khi thuê hay sa th i m t công nhân thì u ph i m t m t chi phí xác nh, m i công nhân c thuê s v n nh n c l ng tháng ngay c khi không s d ng anh ta làm vi c.

V i m i công nhân, ng i qu n lý bi t chi phí thuê, chi phí sa th i và ti n l ng ph i tr cho công nhân ó trong 1 tháng. Và bài toán t ra nh sau: C n ph i thuê hay sa th i bao nhiêu công nhân m i tháng t ng chi phí dành cho nhân công c a d án là nh nh t, t c là gi m t i a chi phí c a d án.

D li u vào: c t file v n b n EMPLOY.IN có c u trúc nh sau:

- Dòng u ghi T là s tháng di n ra d án ($T \leq 100$).

- Dòng thứ hai ghi 3 số nguyên là chi phí thuê, 1 ngày tháng, chi phí sa thải mỗi công nhân.

- Dòng cuối cùng là Tổng nguyên đơn, số thứ tự cho biết số công nhân tối thiểu cần cho tháng thứ i, các giá trị số không quá 150.

Kết quả: Ghi ra file văn bản EMPLOY.OUT theo hình dạng:

- Dòng thứ nhất ghi tổng chi phí nhỏ nhất tìm được.

- Dòng thứ hai ghi Tổng số, số thứ tự là số công nhân hoạt động trong dự án tại tháng thứ i.

Ví dụ: văn file dữ liệu vào và file kết quả ra:

EMPLOY.IN	EMPLOY.OUT
3	265
4 5 6	10 10 11
10 9 11	

* Hướng dẫn:

Bài này hơi phức tạp khó hiểu một chút vì vậy cần phải cẩn thận. Rõ ràng việc thuê thêm hay sa thải công nhân đều phải chi phí tiền nên thay vì muốn biết chi phí ít nhất cho việc thuê nhân công trong các dự án thì số công nhân đang thuê trong một tháng không nhất thiết phải là số công nhân tối thiểu cần cho tháng tiếp theo. Ví dụ, phí tiền thuê cũng như sa thải công nhân rất lớn thì không đáng gì ta lại thường xuyên thuê rồi lại sa thải ngay trong cùng tháng, tốt nhất nên giữ và trả lại (dù họ không làm gì). Nhiệm vụ của chúng ta trong một tháng phải quyết định có bao nhiêu công nhân trong biên chế thuê, nghĩa là phải quyết định xem cần thuê hay cần sa thải bao nhiêu công nhân trong biên chế của tháng trước. Nếu gọi T_{max} là số công nhân của tháng cần nhiều người nhất thì rõ ràng số công nhân trong biên chế thuê của một tháng bất kỳ trong dự án không bao giờ vượt quá T_{max} . Xét một tháng nào đó,

biết rằng không nên sử dụng quá T_{\max} người và cũng không có phép sử dụng ít hơn số người thi đua cho tháng y , nhưng phải chọn giá trị nào trong khoảng giới hạn này. Nếu bạn có các giá trị của T và T_{\max} là như thế thì có thể duy trì tìm ra kết quả tối ưu nhờ do các giá trị này lại có thể khá lớn nên buộc phải tìm cách khác hiệu quả hơn.

Lập mảng $Scn[1..T]$, $Scn[i]$ cho biết số công nhân thi đua cho tháng i , $i=1..T$ (mảng này nhập vào từ file input). Lập thêm mảng $C[T, T_{\max}]$ trong đó $C[i, j]$ cho biết chi phí thi đua của i tháng đầu tiên của đơn vị i tháng thì có j công nhân trong biên chế thuê ($i=1..T$, $j=1..T_{\max}$). Thay là giá trị $C[i, j]$ có thể xác định thông qua các giá trị C từ tháng $i-1$ (là tháng trước):

$$C[i, j] = \text{Min}\{C[i-1, k] + \text{chi phí thuê } k \text{ người thành } j \text{ người}\}$$

$$(i=1..T; j=Scn[i]..T_{\max}; k=Scn[i-1]..T_{\max})$$

Bài toán sẽ giải nhanh hơn khi có các công thức truy hồi và vì phức tạp tính toán không lớn nên chỉ cần chỉ ra trình số ngắn gọn, cho kết quả tối ưu trong thời gian rất ngắn. Kết quả tối ưu cuối cùng là:

$$Kq = \text{Min}\{C[T, j] + \text{chi phí thuê } j \text{ người}\}$$

$$(j=Scn[T]..T_{\max})$$

Có thể tìm ra kết quả ngay (số công nhân sử dụng mỗi tháng) thì cần thêm mảng hai chiều Pre , trong đó $Pre[i, j] := k$, với $i=1..T$; $j=Scn[i]..T_{\max}$; k thoả mãn rằng $(C[i-1, k] + \text{chi phí thuê } k \text{ người thành } j \text{ người})$ nhỏ nhất. Vì chỉ cần tìm kết quả trong mảng Pre không khó.

Bài 17: Cấu hình chèn nhét

Ta cần cấu trúc dữ liệu hình chữ nhật có kích thước $M \times N$ (M, N nguyên dương không lớn hơn 100) thành một số ít nhất các hình vuông có kích thước nguyên dương và có cạnh song song với cạnh của hình chữ nhật ban đầu. Biết rằng khi

cắt mảnh hình chữ nhật bất kỳ thành các mảnh theo 1 phương song song với một trục trong các trục của hình chữ nhật đó.

Dữ liệu vào: một file văn bản CUT.INP có một dòng ghi 2 số M, N .

Kết quả: Ghi ra file CUT.OUT có cấu trúc như sau:

Dòng thứ nhất ghi số K là số hình vuông ít nhất cắt ra.

Dòng thứ 2 ghi kích thước của K hình vuông cắt ra, mỗi số cách nhau 1 dấu cách.

Ví dụ :

CUT.INP	CUT.OUT
5 6	5 2 2 2 3 3

Mô tả trên hình vẽ : (Số ghi trong hình vuông cho biết kích thước của hình vuông đó)

2	2	2
3	3	

***Hỏi:**

+ Nếu $M = 1$ thì cần ít nhất bao nhiêu hình vuông thoả mãn?

+ Nếu $N = 1$ thì cần ít nhất bao nhiêu hình vuông thoả mãn?

+ Nếu gọi $F[i, j]$ là số hình vuông ít nhất cắt ra từ hình chữ nhật kích thước i, j (với $1 \leq i \leq M; 1 \leq j \leq N$) thì công thức QH nào xây dựng nên?

Bài toán con cơ sở : $F[1, j] := j$ với mọi j ; $F[i, 1] := i$ với mọi i

Công thức QH :

Với $i := 2, \dots, m$

Với $j := 2, \dots, n$

Nếu $i = j$ thì $F[i, i] := 1$

Trái l i

$t1 := \min\{F[k,j] + F[i-k, j]\} \forall i k = 1, 2, \dots, i \text{ div } 2;$

$t2 := \min\{F[i,k] + F[i, j-k]\} \forall i k = 1, 2, \dots, j \text{ div } 2;$

$F[i,j] := \min\{t1, t2\}$

***Chương trình cài đặt:**

```
program cut_hcn;
const fi='cut.inp';
      fo='cut.out';
      maxn=1001;
var   f:text;
      m,n,t:integer;
      q:array[1..maxn,1..maxn] of integer;
procedure   nhap;
var   i,j:integer;
begin
    assign(f,fi);
    reset(f);
    readln(f,m,n);
    for i:=1 to m do
        for j:=1 to n do
            q[i,j]:=-1;
    close(f);
end;
procedure   xuli;
var min,l,k,i,j:integer ;
begin
```

```

for i:=1 to m do q[i,1]:=i;
for i:=1 to n do q[1,i]:=i;
for i:=2 to m do
    for j:=2 to n do
        if i=j then q[i,j]:=1
        else
            begin
                min:=maxint;
                for k:=1 to j div 2 do
                    if min>q[i,k]+q[i,j-k] then min:=q[i,k]+q[i,j-k] ;
                for l:=1 to i div 2 do
                    if min>q[l,j]+q[i-l,j] then min:=q[l,j]+q[i-l,j] ;
                q[i,j]:=min;
            end;
        {if i=-1 then
        if (i mod j =0) or (j mod i =0) then
            begin
                if i>=j then q[i,j]:=i div j else q[i,j]:=j div i;
            end
        else
            begin
                min:=maxint;
                for k:=1 to j div 2 do
                    if min>tinhq(i,k)+tinhq(i,j-k) then min:=tinhq(i,k)+tinhq(i,j-
k) ;
                for l:=1 to i div 2 do

```

```

        if min>tinh q(l,j)+tinhq(i-l,j) then
            min:=tinhq(l,j)+tinhq(i-l,j) ;
        q[i,j]:=min;
    end;
    tinhq:=q[i,j]; }
end;
procedure    timkq(i,j:integer);
var k,l,tg:integer;
begin
    if (i mod j =0) or (j mod i =0) then
        begin
            if i>=j then for tg:=1 to i div j do write(f,j,' ')
            else for tg:=1 to j div i do write(f,i,' ');
        end
    else
        begin
            for k:=1 to (j div 2) do
                if q[i,j]=(q[i,k]+q[i,j-k]) then
                    begin
                        timkq(i,k);
                        timkq(i,j-k);
                        exit;
                    end ;
            for l:=1 to (i div 2) do
                if q[i,j]=(q[l,j]+q[i-l,j]) then
                    begin

```

```

        timkq(l,j);
        timkq(i-l,j);
        exit;
    end;
end;
end;
procedure   xuất;
var j,i:integer;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,q[m,n]);
    timkq(m,n);
    close(f)
end;
BEGIN
    nhap;
    xuli;
    xuất;
END.

```

Bài 18: Trò chơi v i b n g s : Rate This

Trò chơi v i b n g s là trò chơi tham gia trùng th ng c mô t nh sau: Có m t b n g hình ch nh t c chia ra làm n ô vuông, ánh s t trái qua ph i b t u t 1. Trên ô vuông th i ng i ta ghi m t s nguyên d ng a_i , $i = 1, 2, \dots, n$. m t l t ch i, ng i tham gia trò chơi c quy n l a ch n m t s l ng tùy ý các ô trên b n g s . Gi s theo th t t trái qua ph i,

ng i ch i l a ch n các ô i_1, i_2, \dots, i_k . Khi ó i m s mà ng i ch i t c s là:

$$a_{i_1} - a_{i_2} + \dots + (-1)^{k-1} a_{i_k}$$

Yêu c u: Hãy tính s i m l n nh t có th t c t m t l t ch i.

D li u vào: c t file v n b n Rate.inp có c u trúc:

Dòng u tiên ch a s nguyên d ng n ($n \leq 10^6$) là s l ng ô c a b ng s ;

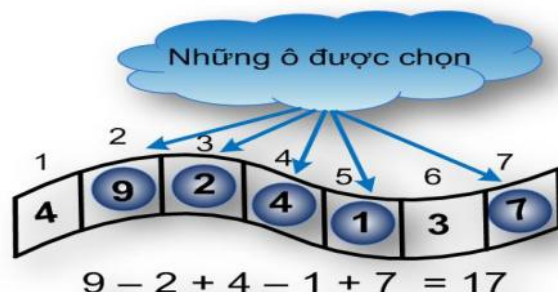
Dòng th hai ch a n s nguyên d ng a_1, a_2, \dots, a_n ($a_i \leq 10^4, i = 1, 2, \dots, n$) ghi trên b ng s . Các s liên ti p trên cùng dòng c ghi cách nhau b i ít nh t m t d u cách.

K t qu ra: Ghi ra file v n b n Rate.out có c u trúc là:

M t s nguyên duy nh t là s i m l n nh t có th t c t m t l t ch i.

Ví d :

Rate.inp	Rate.out
7	17
4 9 2 4 1 3 7	



Ràng bu c: 60% s tests ng v i 60% s i m c a bài có $1 \leq n \leq 20$.

***Ch ng trình cài t:**

```
program rate_this;
```

```
const fi='ratethis.inp';
```

```

    fo='ratethis.out';
    maxn=1000000;
var   f:text;
    n,kq,dem:int64;
    a,fl,fc,ds:array[1..maxn] of int64;
    luuc,luul:array[1..maxn] of integer;
procedure   nhap;
var   i:longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        read(f,a[i]) ;
    close(f);
    dem:=0;
    fillchar(luul,sizeof(luul),0);
    fillchar(luuc,sizeof(luuc),0);

end;
function   max2so(cl,j,x,y:int64):int64;
begin

    if x>y then  max2so:=x
    else  begin
        max2so:=y;

```

```

        if cl=0 then luuc[j]:=1;
        if cl=1 then luul[j]:=1;
    end;

end;

procedure    xuli;
var i,chanle:longint;
begin
    fl[1]:=a[1];
    fc[1]:=0;
    for i:=2 to n do
        begin
            fl[i]:=max2so(1,i,fl[i-1],fc[i-1]+a[i]);
            fc[i]:=max2so(0,i,fc[i-1],fl[i-1]-a[i]);
        end;
    if fl[n]>fc[n] then begin kq:=fl[n];chanle:=1;end
    else begin kq:=fc[n];chanle:=0;end;
    for i:=n downto 1 do
        begin
            if (chanle=1) and (luul[i]=1) then
                begin
                    inc(dem);
                    ds[dem]:=i;
                    chanle:=0;
                    continue;
                end;
        end;

```



```

        if (chanle=0) and (luuc[i]=1) then
            begin
                inc(dem);
                ds[dem]:=i;
                chanle:=1;
            end;
        end;

end;

procedure    xuat;
var i:longint;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,kq);
    for i:=dem downto 1 do
        write(f,ds[i],' ');
    close(f)
end;

BEGIN
    nhap;
    xuli;
    xuat;
END.

```

Bài 19: Con ki n

Trên m t sân hình ch nh t $M \times N$, c chia thành các ô vuông n v , m i ô ch a m t l ng th c n. M t con ki n xu t phát t ô (1,1) mu n i qua sân n dòng th M. Con ki n ch có th i theo m t dòng chia nh trên sân ng v i m t dòng c a b ng ch nh tho c i theo trên m t c t c a sân. Hãy ch ra ng i giúp con ki n có c nhi u th c n nh t.

D li u vào: c t file v n b n FOOD.INP có c u trúc:

Dòng u là 2 s M, N.

M dòng ti p theo, m i dòng có N s t ng ng là l ng th c n có trong $M \times N$ ô trên sân.

K t qu ra: Ghi ra file v n b n FOOD.OUT có c u trúc:

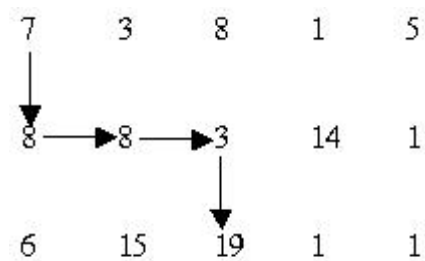
Dòng u là t ng l ng th c n nhi u nh t mà con ki n có th n c.

Các dòng ti p theo, m i dòng là 2 s t ng ng là to dòng, c t c a ô mà con ki n i qua.

Ví d :

FOOD.INP

3 5



FOOD.OUT

45

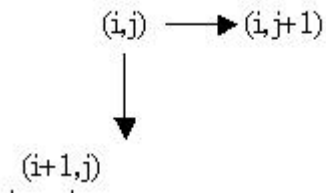
```

1 1
2 1
2 2
2 3
  
```

***Hạng d n:**

Trng s này là l ng th c n trên m i ô.

Quy t c i:



D dàng tìm c công th c quy ho ch ng là:

G i $B[i,j]$ là l ng th c n l n nh t i t ô $(1,1)$ n ô (i,j)

$B[1,j] = A[1,j]$ v i $j = 1..N$

$B[i,1] = A[i,1] + B[i-1,1]$ v i $i = 2..M$

$B[i,j] = \max\{B[i-1,j], B[i,j-1]\} + A[i,j]$ v i $i = 2..M$ và $j = 2..N$

Bài 20: Sa m c

M t bãi sa m c có d ng hình ch nh t $M \times N$. M i ô vuông n v trên sa m c có m t cao nào ó. M t ng i mu n i t b u này sang b cu i cùng bên kia. Ng i ó ch có th i t ô ang ng t i m tô m i theo h ng th ng ng chéo trái ho c chéo ph i. Gi thi t r ng ng i ó không c v t ra hai mép trái và ph i c a sa m c.

Hãy tìm ng i sao cho ng i ó ph i v t qua quăng ng ng n nh t. M i l n i t m t ô sang ô m i ti p theo ng i ó ph i i h t quăng ng b ng chênh cao gi a hai ô ó.

SAMAC.INP

5	5			
3	3	8	1	5
8	7	3	14	1
6	7	18	1	1
20	20	17	23	24
31	20	27	10	6

SAMAC.OUT

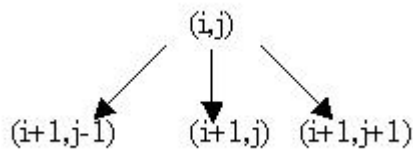
12(Quảng ng Min)

(1,3)(2,4) (3,3) (4,2) (5,2)

***H ng d n:**

Tr ng s là chênh cao gi a hai ô liên ti p.

Quy t c i.



Công th c t i u:

$B[i,j]$ là quãng ng nh nh t i t b u tiên n ô (i,j) .

$B[1,j] = 0 \forall j = 1..N$

$B[i,1] = \text{Min} \{ B[i-1,1] + \text{abs}(A[i,1] - A[i-1,1]), B[i-1,2] + \text{abs}(A[i,1] - A[i-1,2]) \}$

$\forall i = 2..M$

$B[i,j] = \text{Min} \{ B[i-1,j-1] + \text{abs}(A[i,j] - A[i-1,j-1]), B[i-1,j] + \text{abs}(A[i,j] - A[i-1,j]),$

$B[i-1,j+1] + \text{abs}(A[i,j] - A[i-1,j+1]) \} \forall i = 2..M, j = 2..N-1$

$B[i,N] = \text{Min} \{ B[i-1,N] + \text{abs}(A[i,N] - A[i-1,N]), B[i-1,N-1] + \text{abs}(A[i,N] - A[i-$

$1,N-1]) \}, \forall i = 2..M.$

Bài 21: Quy bán hàng.

Một siêu thị có M gian hàng, mỗi gian hàng gồm N ng n ch a, mỗi ng n ch a có b trí m t phòng. Giám c siêu thị quy t nh m m t t khuy n mãi cho khách hàng v i các quy t c sau:

M i gian hàng c b trí trên t ng t ng t ng ng t t ng l n M. M i t ng có N thang máy i lên ng v i m i phòng.

M t khách hàng có th mua s n ph m t i m t gian hàng nh ng ch có th i theo m t h ng (không c mua xong r i quay tr l i n i ã mua). Khách hàng có th i thang máy lên t ng ti p theo, nh ng ph i mua ít nh t t i m t ng n ch a t ng ó thì m i c phép i lên t ng trên n a.

Khách hàng mua hàng t i m t ng n ch a. M i ng n ch a quy nh m t s l ng hàng mà ng i khách bu c ph i mua khi n ng n ch a ó. N u chênh s l ng hàng gi a hai ng n ch a liên ti p c a m t khách hàng là m t s may m n ã bi t tr c. Khách hàng ó s c khuy n mãi thêm m t s hàng b ng chính s may m n ó.

n t ng th M, khách hàng ch có th mua hàng t i duy nh t m t ng n ch a.

Hãy giúp khách hàng l a ch n i m xu t phát và h ng i sao cho mua c nhi u hàng nh t (K c s hàng c khuy n mãi).

D li u u vào cho trong FILE v n b n SHOP.INP có c u trúc nh sau:

Dòng u là 3 s M, N, K ($1 < M, N, K \leq 100$), K là s các con s may m n.

Dòng th hai ghi K con s may m n.

M dòng ti p theo ghi s l ng hàng quy nh t i m i ng n ch a. M i dòng g m N s cách nhau b i ít nh t m t d u tr ng.

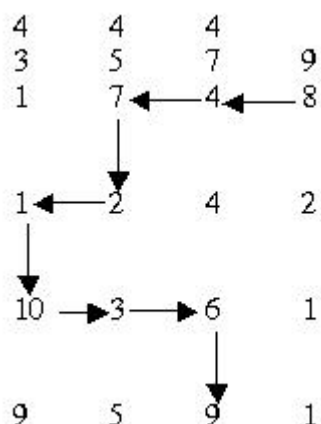
K t qu ghi ra FILE v n b n SHOP.OUT nh sau:

Dòng m t là s l ng hàng nhi u nh t.

Dòng hai i m xu t phát và quá trình mua hàng. M i ng n ch a i qua c bi u di n theo d ng (x,y) trong ó (x,y) là v trí c a ng n ch a.

Ví d :

SHOP.INP



SHOP.OUT

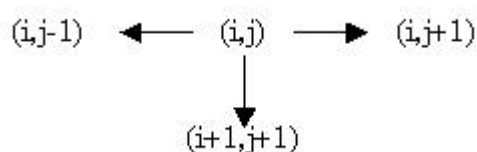
80 (L ng hàng mua c Max)

(1,3) (1,2) (2,2) (2,1) (3,1) (3,2) (3,3) (3,4) (4,4)

***H ng d n:**

Tr ng s ây là s l ng hàng t i các ng n ch a. ng th i khi tho măn i u ki n khuy n mãi thì tr ng s s c t ng thêm s l ng b ng s các con s may m n (ph thu c vào ô ng tr c).

Quy t c i:



:

Công th c quy ho ch ng:

$B[i,j]$ là l ng hàng max khi i t t ng 1 cho n ng n ch a (i,j)

$B[0,j] = 0$ v i $j = 1..N$

$B[i,0] = 0$ v i $i = 1..M$

$B[i,j] = \text{Max}\{B[i,j-1]+KM1, B[i-1,j] + KM2, B[i-1,k]+SA[i,u]+KM3\} + A[i,j]$. Với
 $i = 1..M$, $j = 1..(N-1)$, $k = (j+1)..M$, $u = j+1..k$
 KM1 là 1 hàng khuy n mỗi n u $\text{Abs}(A[i,j]-A[i,j-1])$ là con s may m n.
 KM2 là 1 hàng khuy n mỗi n u $\text{Abs}(A[i,j]-A[i-1,j])$ là con s may m n.
 KM3 là t ng s 1 hàng khuy n mỗi n u $\text{Abs}(A[i,k]-A[i-1,k])$ và $\text{Abs}(A[i,t]-A[i,t+1])$ v i $t = j..(u-1)$ là các con s may m n.
 $B[i,N] = \text{Max}\{B[i,N-1]+KM1, B[i-1,N]+KM2\} + A[i,N]$

V i $i = 1..M$

KM1 là 1 hàng khuy n mỗi n u $\text{abs}(A[i,N]-A[i,N-1])$ là con s may m n.

KM2 là 1 hàng khuy n mỗi n u $\text{abs}(A[i,N]-A[i-1,N])$ là con s may m n.

Nh n xét:

Còn r t nhi u bài toán khác có d ng nh m t trong các bài toán c b n này nh ng chung quy l i chúng ta u có th a nó v m t d ng chung. Sau ó đ a vào nh ng nguyên t c gi i chung, ta u có th gi i quy t đ d dàng. Các d ng bài toán t ng quát này khi đ li u cho quá gi i h n khai báo b ng hai chi u u có th gi i quy t b ng cách quy ho ch liên t c trên 2m ng m t chi u. Sau m i b c quy ho ch ph i thay i 2 m ng này sao cho phù h p v i b c quy ho ch ti p theo. Cái khó c a bài toán có đ li u l n này là v i c l u tr tr ng thái sau khi quy ho ch toàn b ta còn có th in ra file k t qu quá trình i c a ph ng án t i u.

Chương IV. Một số thuật giải

Bài 22: Chia thành nhiều nhóm có tổng bằng nhau

Cho dãy a_1, a_2, \dots, a_n ($n \in \mathbb{N}^*$). Tìm cách chia dãy trên thành m nhóm sao cho mỗi nhóm có tổng bằng nhau và tìm kiếm là lớn nhất.

Dòng vào: Đọc vào tệp văn bản MNHOM.INP có cấu trúc như sau:

Dòng đầu ghi số nguyên dương ($N \leq 10^4$)

Các dòng tiếp theo ghi dãy a_1, a_2, \dots, a_n , các số cách nhau ít nhất 1 dấu cách;

Kết quả: Ghi ra tệp văn bản: MNHOM.OUT có cấu trúc như sau:

Dòng đầu tiên ghi số lớn nhất tìm được

M dòng tiếp theo, mỗi dòng ghi các chỉ số của các phần tử thuộc cùng 1 nhóm;

Ví dụ:

MNHOM.INP	MNHOM.OUT
7	4
8 2 1 3 9 10 11	1 4
	2 5
	3 6
	7

Bài 23: Bài toán xoay DOMINO

Cho N thanh DOMINO xếp theo chiều dọc như hình vẽ.

1	1	3	2	6
2	6	5	6	4

Ví dụ hình trên gồm 5 thanh DOMINO.

M i thanh DOMINO g m 2 ph n, ph n trên và ph n d i. Trên m i ph n có m t s t l n 6. Yêu c u t ra là hãy tìm cách xoay các thanh (xoay 180) sau khi xoay chênh l ch gi a t ng trên và t ng d i là ít nh t. *Gi i h n:* $N \leq 1000$.

D li u vào: c t file v n b n DOMINO.INP có c u trúc nh sau:

- Dòng u tiên ghi s nguyên d ng N
- N dòng ti p theo, m i dòng i ghi 2 s nguyên a_i, b_i ($1 \leq a_i, b_i \leq 6$) t ng ng là s trên, d i c a thanh DOMINO th i;

K t qu ra: Ghi ra file v n b n DOMINO.OUT có c u trúc nh sau:

- Dòng u tiên ghi giá tr chênh l ch gi a t ng trên và t ng i;
- Các dòng ti p theo là ch s c a các thanh DOMINO c xoay;

Ví d :

DOMINO.INP	DOMINO.OUT
5	0
1 2	1
1 6	
3 5	
2 6	
6 4	

Bài 24: Bài toán ngân hàng tr ti n;

M t ng i i l y ti n m t ngân hàng. Anh ta c n l y m t kho n úng M ng. Ngân hàng có N ng ti n A_1, A_2, \dots, A_N . H i ngân hàng có bao nhiêu cách tr ti n.

D li u vào: c t file v n b n "MONEY.INP" có d ng:

Dòng u là hai s N và M ($N \leq 100, M \leq 10000$)

Các dòng ti p theo là các ph n t c a m ng A .

K t qu ra: Ghi ra file v n b n "MONEY.OUT" g m m t dòng duy nh t là s cách tr ti n (S cách tr ti n < Maxlongint)

Ví d :

MONEY.INP	MONEY.OUT
5 10 1 2 3 4 5	3

Bài 25: Bài toán dãy có t ng chia h t cho k (thi toàn Qu c)

Cho m t dãy s nguyên A_1, A_2, \dots, A_N và m t s k. Hãy tìm m t dãy con (không nh t thi t ph i liên ti p nhau) dài nh t có t ng các s chia h t cho s k.

D li u vào: c t file v n b n "dayso.inp" có d ng:

Dòng 1 g m 2 s N và k ($N \leq 1000$; $k \leq 50$)

Các dòng ti p theo ch a các s c a m ng A.

K t qu ra: Ghi ra file v n b n "dayso.out" g m m t dòng ghi s ph n t l n nh t tìm c.

Ví d :

DAYSO.INP	DAYSO.OUT
6 5 1 2 7 3 4 5	5

Bài 26: X p v t vào ba lô (m i lo i v t có lý ng không h n ch);

M t chi c ba lô có th ch a c m t kh i l ng không quá W. Có N lo i v t c ánh s t l n N, m i v t lo i i có kh i A_i và giá tr s d ng C_i ($i=1, 2, \dots, N$), s l ng v t m i lo i là không h n ch . Hãy ch n các v t (m i lo i bao nhiêu cái) x p vào ba lô sao cho t ng giá tr các v t trong ba lô là l n nh t (n u có th x p c).

D li u vào: c t file v n b n BALO.INP có c u trúc nh sau:

Dòng u ghi 2 s nguyên d ng N, W ($0 < N, W \leq 100$).

Dòng $i+1$, $1 \leq i \leq N$) ghi 2 s nguyên d ng A_i và C_i ($0 < A_i, C_i < 256$)

Kết quả: Ghi ra file văn bản BALO.OUT có cấu trúc như sau:

Dòng ghi từng giá trị liên tiếp tìm được.

Từ dòng thứ 2, mỗi dòng ghi 2 số là: số hiện tại và số liền kề hiện tại.

Ví dụ :

BALO.INP	BALO.OUT
9 100	965
60 7	5 10
8 45	9 3
50 15	
9 7	
7 68	
130 57	
1 3	
70 123	
10 95	

Bài 27: Tìm kiếm

Tên của Omega đang là tiêu chuẩn xu. Có N loại tiêu chuẩn xu, loại thứ i có mệnh giá là a_i đồng. Một ngày khách du lịch đến Omega du lịch và sẽ tiêu tiền. Ông ta muốn biết số tiền tối thiểu để mua được N loại tiêu chuẩn xu. Ông ta cũng muốn biết số tiền tối thiểu để mua được N loại tiêu chuẩn xu (cho tất cả các loại tiêu chuẩn xu). Bạn hãy giúp ông ta tìm cách tìm kiếm.

Dữ liệu: vào trong file văn bản Money.inp có dạng:

- Dòng đầu là 2 số N và M ($N \leq 100$ và $M \leq 10000$)
- Các dòng tiếp theo là N số nguyên là các mệnh giá a_1, a_2, \dots, a_n ($\forall i, 0 < a_i \leq 1000$)

K t qu : Ghi ra file v n b n Money.out g m m t dòng duy nh t là s ng ti n ít nh t có th i c (S ng xu < Maxlongint)

Ví d :

Money.inp	Money.out
4 10 1 2 3 4	3
Money.inp	Money.out
5 10 2 4 6 8 9	2

Bài 28: Chia t p

Xét t p ch a các s nguyên t 1 t i N v i $N = 4K - 1$ ho c $N = 4K$. T p các s nguyên này có th chia thành hai t p con có t ng các ph n t b ng nhau. Ví d , v i $N = 7$, ta có 4 cách chia tho mẫn i u ki n trên:

{2,3,4,5} và {1,6,7}

{1,3,4,6} và {2,5,7}

{1,2,5,6} và {3,4,7}

{1,2,4,7} và {3,5,6}

V i N cho tr c, hãy tính s cách chia thành 2 t p con có t ng các ph n t b ng nhau.

D li u: Vào t file v n b n SET.INP, m i dòng ch a m t s nguyên N ($N \leq 100$);

K t qu : a ra file v n b n SET.OUT s cách chia, m i dòng ng v i 1 dòng c a file d li u vào

Ví d :

SET.INP	SET.OUT
7	4

11	35
----	----

Bài 29: i t i n xu

N c Silverland s d ng h th ng t i n xu, trong ó các xu có m nh giá là m t s chính ph ng: 1,4,9,...,289(=17²). V i h th ng này, tr 10 xu ta có 4 cách:

Tr 10 ng 1 xu,

Tr 1 ng 4 xu và 6 ng 1 xu,

Tr 2 ng 4 xu và 2 ng 1 xu,

Tr 1 ng 9 xu và 1 ng 1 xu.

Nhi m v c a b n là xác nh xem có bao nhiêu cách tr m t s t i n cho tr c n c Silverland.

D li u vào: Vào t file v n b n COIN.INP

G m nhi u dòng, m i dòng m t s nguyên d ng không v t quá 800.

K t qu ra: a ra file v n b n COIN.OUT là s cách tr ng v i t ng tr ng h p.

Ví d :

COIN.INP
2
10
30

COIN.OUT
1
4

M C L C

I.1. T t ng c a ph ng pháp quy ho ch ng.....	3
I.1.1. Thu t toán chia tr	3
I.1.2. H th c truy h i	4
I.1.3. L p trình ng là gì?.....	5
I.1.4. Ph ng pháp quy ho ch ng	7
I.2. Các b c th c hi n gi i bài toán b ng ph ng pháp quy ho ch ng	8
I.2.1. Các b c c b n:	8
I.2.2. T ch c cài t:	9
I.2.3 Khi nào dùng ph ng pháp quy ho ch ng?	9
Bài 1: Tìm dãy con không gi m nhi u ph n t nh t;.....	10
H ng d n:.....	10
* Ch ng trình cài t:.....	11
Bài 2: Dãy con chung dài nh t;.....	13
Víd :	14
*H ng d n:.....	14
*Ch ng trình cài t:.....	15
Bài 3: Dãy con có t ng b ng S	17
Bài 4: X p v t vào ba lô (<i>m i v t ch có 1</i>)	20
*H ng d n.....	21
*Ch ng trình cài t:.....	22
Ch ng III: Bài t p ch n l c	26
Bài 5: B trí phòng h p;	26
* H ng d n:.....	27
* Ch ng trình cài t:.....	27
Bài 6: Cho thuê máy tính;	30
*H ng d n:.....	31
* Ch ng trình cài t:.....	31
Bài 7: N i i m.....	34
*H ng d n:.....	35
* Ch ng trình cài t:.....	35
Bài 8: Dãy con i chi u, i d u dài nh t;.....	38
* H ng d n:.....	39
* Ch ng trình cài t:.....	39
Bài 9: S phép bi n i ít nh t.....	43
* H ng d n:.....	43
*Ch ng trình cài t:.....	44
Bài 10: Xâu i x ng;.....	47
*H ng d n:.....	47
*Ch ng trình cài t:.....	47
Bài 11: Bài toán chia k o.....	50

*Hạng d n:	50
*Chương trình cài t:	50
Bài 12: Mua cá theo lý ng (<i>Olympic Balkan 2000</i>) ;	54
*Hạng d n:	55
*Chương trình cài t:	55
Bài 13: i n d u h i vào bi u th c	57
*Hạng d n:	57
*Chương trình cài t:	58
Bài 14: Chia thành hai nhóm có tích l n nh t (<i>ACM 10690</i>)	61
*Hạng d n:	62
*Chương trình cài t:	62
Bài 15: Bài toán mua bán hàng	64
*Chương trình cài t:	65
Bài 16: L ch thuê nhân công	70
*Hạng d n:	71
Bài 17: C t hình ch nh t	72
*Hạng d n:	73
*Chương trình cài t:	74
Bài 18: Trò ch i v i b ng s : Rate This	77
*Chương trình cài t:	78
Bài 19: Con ki n	82
*Hạng d n:	83
Bài 20: Sa m c	83
*Hạng d n:	84
Bài 21: Qu y bán hàng.	85
*Hạng d n:	86
Nh n xét:	87
Bài 22: Chia thành nhi u nhóm có t ng b ng nhau	88
Bài 23: Bài toán xoay DOMINO	88
Bài 24: Bài toán ngân hàng tr t i n;	89
Bài 25: Bài toán dãy có t ng chia h t cho k (thi toàn Qu c)	90
Bài 26: X p v t vào ba lô (<i>m i lo i v t có lý ng không h n ch</i>);	90
Bài 27: i t i n	91
Bài 28: Chia t p	92
Bài 29: i t i n xu	93

Tài li u tham kh o

1. Tr n Hùng, *Chuyên b i d ng HSG Tin h c THPT*, NXB GD 2007
2. Nguy n Quý Khang, *Bài t p Pascal*, NXB QGHN 2002
3. Nguy n Xuân My, *M t s v n ch n l c trong môn Tin h c*, NXB GD 2002
4. Nguy n Xuân My, *Bài t p l p trình Pascal*, NXB TK 1997
5. Báo TH&NT 1999 - 2006
6. Trang web: <http://www.wikipedia.org>
7. Trang web: <http://www.vnoi.info>
8. Trang web: <http://www.ddth.com>