**DÃY SỐ HÌNH NÓN** *Tên chương trình*: BITONIC.???

Dãy số ***b***1, ***b***2, . . ., ***bm*** được gọi là hình nón nếu tồn tại ***j*** sao cho ***b1*** < ***b***2 < . . .< ***bj*** > ***bj***+1 > . . . > ***bm*** (1 <***j***<***m***). Theo định nghĩa ta thấy dãy số hình nón phải có ít nhất 3 phần tử.

Cho dãy số nguyên ***a***1, ***a***2, . . ., ***an***. Dãy số ***ai***1, ***ai***2, . . ., ***aik*** được gọi là dãy con của dãy đã cho nếu 1 ≤ ***i***1 < ***i***2 < . . . < ***ik*** ≤ ***n***.

***Yêu cầu***: Cho dãy số nguyên, các số ở hệ thập phân. Hãy xác định dãy con hình nón có tổng các chữ số lớn nhất và đưa ra tổng này.

***Dữ liệu***: Vào từ file văn bản BITONIC.INP:

* Dòng đầu tiên chứa số nguyên ***n*** (3 ≤ ***n*** ≤ 1 000),
* Dòng thứ 2 chứa ***n*** số nguyên ***a***1, ***a***2, . . ., ***an*** (1 ≤ ***ai*** ≤ 105).

***Kết quả***: Đưa ra file văn bản BITONIC.OUT một số nguyên – kết quả tìm được.

***Ví dụ***:

|  |  |  |
| --- | --- | --- |
| BITONIC.INP |  | BITONIC.OUT |
| **4**  **2 1 3 1** |  | **6** |
|  |

Nối mạng máy tính

Các học sinh khi đến thực tập trong phòng máy tính thường hay chơi trò chơi điện tử trên mạng. Để ngăn ngừa, người trực phòng máy đã ngắt tất cả các máy tính ra khỏi mạng và xếp chúng thành một dãy trên một cái bàn dài và gắn chặt máy xuống mặt bàn rồi đánh số thứ tự các máy từ 1 đến N theo chiều từ trái sang phải. Các học sinh tinh nghịch không chịu thua, họ đã quyết định tìm cách nối các máy trên bàn bởi các đoạn dây nối sao cho *mỗi máy được nối với ít nhất một máy khác*. Để tiến hành công việc này, họ đã đo khoảng cách giữa hai máy liên tiếp. Bạn hãy giúp các học sinh này tìm cách nối mạng thoả mãn yêu cầu đặt ra sao cho tổng độ dài cáp nối phải sử dụng là ít nhất.

**Dữ liệu:** vào từ file văn bản CABLE.INP:

* Dòng đầu tiên chứa số lượng máy N (1≤N≤25000)
* Dòng thứ i trong số N-1 dòng tiếp theo chứa các khoảng cách từ máy i đến máy i+1 (i=1,2,...,N-1). Giả thiết rằng khoảng cách từ máy 1 đến máy N không vượt quá 106.

**Kết quả:** Ghi ra file văn bản CABLE.OUT độ dài của cáp nối cần sử dụng.

*Ví dụ:*

|  |  |
| --- | --- |
| CABLE.INP | CABLE.OUT |
| 6  2  2  3  2  2 | 7 |

Hướng dẫn:

Lưu mảng A[1] = 0, A[i] = khoảng cách từ i-1 tới i.

Gọi F[i] là độ dài đoạn dây ngắn nhất để nối các máy từ máy 1 tới máy i

F[1] = 0;

F[2] = A[2];

F[3] = F[2] + A[3];

với i >=4: thì máy i sẽ nối với máy i – 1.

* Nếu máy i – 1 không nối với máy i – 2 thì: F[i] = F[i-2] + A[i]
* Nếu máy i – 1 nối với máy i – 2 thì F[i] = F[i-1] + A[i]

Do cần tìm đoạn dây ngắn nhất nên: F[i] = Min(F[i-1], F[i-2]) + A[i]

# Đoàn xe qua cầu

Cho một đoàn xe gồm n chiếc đi trên một đường một chiều và đoàn xe đã được bố trí theo thứ tự từ 1 đến n. Mỗi một xe trong đoàn có vận tốc là ***Vi*** và trọng lượng ***Wi***.

Khi đi qua một chiếc cầu có trọng tải giới hạn là P thì đoàn xe phải chia thành các nhóm sao cho tổng trọng lượng của mỗi nhóm không quá P (Lưu ý rằng không được đảo thứ tự đoàn xe). Các nhóm phải đi tuần tự có nghĩa là nhóm thứ i chỉ được khởi hành khi mà toàn bộ xe của nhóm thứ i - 1 đã qua cầu. Giả thiết rằng P > *Wi* với ∀i: 1 ≤ i ≤ n.

Rõ ràng khi đó thời gian để một nhóm xe qua cầu phụ thuộc vào xe chậm nhất trong nhóm đó nếu coi như chiều dài cũng như khoảng cách của các xe là không đáng kể.

***Hãy tìm cách chia đoàn xe thành các nhóm sao cho thời gian mà đoàn xe sang được cầu là nhỏ nhất có thể được.***

**Dữ liệu: Vào từ file văn bản CARGROUP.INP**

* Dòng đầu là 3 số nguyên dương n, P và L (n, P, L ≤ 1000) thể hiện cho số xe, trọng lượng giới hạn của cầu và độ dài của cầu.
* Dòng thứ i trong n dòng kế tiếp gồm 2 số nguyên dương ***Wi*** và **V*i*** (***Wi***, ***Vi*** ≤ 100)

**Kết quả: Ghi ra file văn bản CARGROUP.OUT**

* Dòng đầu ghi một số thực là tổng thời gian nhỏ nhất để xe qua cầu, cho phép làm tròn lấy 2 chữ số sau dấu chấm thập phân.
* Dòng kế tiếp gồm các số x1, x2, .., xk thể hiện: nhóm 1 gồm các xe từ 1 đến xe thứ x1, nhóm 2 gồm các xe thứ x1 + 1 đến xe thứ x2..., nhóm k từ xe thứ x[k - 1] tới x[k]

***Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách.***

**Ví dụ:**



|  |  |  |
| --- | --- | --- |
| CARGROUP.INP |  | CARGROUP.OUT |
| 10 100 100  40 25  50 20  50 20  70 10  12 50  09 70  49 30  38 25  27 50  19 70 |  | 25.00  1 3 6 8 10 |

Hương dẫn giải:

Gọi F[i] là thời gian nhỏ nhất để i xe đầu tiên qua cầu.

F[1] = L/W[1]

Xét i > 2:

* Nếu xe i không ghép với đoàn xe nào thì F[i] = F[i-1] + L/W[i]
* Nếu xe i được ghép với một đoàn xe nào đó thì điều kiện là tổng trọng lượng của đoàn xe đó phải <= P
  + Nếu xe i và xe i-1 ghép với nhau thành 1 đoàn xe thì:

F[i] := F[i-2] + L/ Min{W[i],W[i-1]}

* + Nếu xe i và i-1, i-2 ghép thành 1 đoàn xe thì:

F[i] := F[i-3] + L/ Min{W[i],W[i-1],W[i-2]}

* ......
* Nếu xe i và i-1, i-2, i-3, i-4, ..., i – j ghép thành 1 đoàn xe thì

F[i] := F[i-j-1] + L/ Min{W[i],W[i-1],W[i-2],...,W[i-j]}

Vì cần tim thời gian là nhỏ nhất để qua cầu nên

F[i] = Min(F[i-1] +L/W[i], F[i-2] +L/ min{W[i], W[i-1]}, ..., F[i-j-1] + L/Min{W[i],W[i-1],W[i-2],...,W[i-j]})

**Trò chơi với băng số** *Tên chương trình: LINEGAME.PAS*

Trò chơi với băng số là trò chơi tham gia trúng thưởng được mô tả như sau: Có một băng hình chữ nhật được chia ra làm *n* ô vuông, đánh số từ trái qua phải bắt đầu từ 1. Trên ô vuông thứ *i* người ta ghi một số nguyên dương *ai*, *i* = 1, 2, ..., n. Ở một lượt chơi, người tham gia trò chơi được quyền lựa chọn một số lượng tùy ý các ô trên băng số. Giả sử theo thứ tự từ trái qua phải, người chơi lựa chọn các ô *i1*, *i2*, ..., *ik*. Khi đó điểm số mà người chơi đạt được sẽ là:

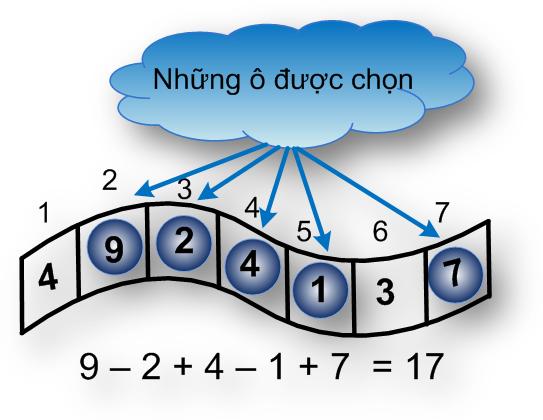
* ai1 - ai2 + ... + (-1)k-1aik

**Yêu cầu:** Hãy tính số điểm lớn nhất có thể đạt được từ một lượt chơi.

**Dữ liệu:** Đọc từ file **LINEGAME.INP**:

* Dòng đầu tiên chứa số nguyên dương n ( n ≤ 106 ) là số lượng ô của băng số;
* Dòng thứ hai chứa n số nguyên dương a1, a2, ..., an ( ai ≤ 104, i = 1, 2, ..., n ) ghi trên băng số. Các số liên tiếp trên cùng dòng được ghi cách nhau bởi ít nhất một dấu cách.

**Kết quả:** Ghi ra file **LINEGAME.OUT**: là số điểm cao nhất đạt được.

**Ví dụ:**

|  |  |
| --- | --- |
| **LINEGAME.INP** | **LINEGAME.OUT** |
| 7 4 9 2 4 1 3 7 | 17 |

**Dãy con không liền kề dài nhất** *Tên chương trình LNACS.PAS*

Dãy C = c1, c2, ..., ck là dãy con không liền kề của dãy A = a1, a2, ..., am nếu C có thể nhận được bằng cách chọn một dãy các phần tử không liền kề của A, nghĩa là tìm dược dãy các chỉ số i1, i2, ..., ik sao cho:

1 ≤ i1, i2, ..., ik ≤ m;

i1 < i2 - 1, i2 < i3 - 1, ..., ik - 1 < ik - 1;

c1 = ai1, c2 = ai2, ck = aik.

Ta gọi độ dài của dãy số là số phần tử của nó.

Cho hai dãy:

A = a1, a2, ..., am và B = b1, b2, ..., bn

Dãy C được gọi là dãy con chung không liền kề của hai dãy A và B nếu như nó vừa là dãy con không liền kề của A, vừa là dãy con không liền kề của B.

**Yêu cầu:** Cho hai dãy số A và B. Hãy tìm độ dài của dãy con chung không liền kề dài nhất của hai dãy đã cho.

**Dữ liệu:** Đọc từ file **LNACS.INP:**

* Dòng đầu tiên chứa hai số nguyên dương m và n (2 ≤ m, n ≤ 103) được ghi cách nhau bởi dấu cách, lần lượt là số lượng phần tử của dãy A và dãy B.
* Dòng thứ i trong m dòng tiếp theo chứa số nguyên không âm ai (ai ≤ 104), i = 1, 2, ..., m.
* Dòng thứ j trong n dòng tiếp theo chứa số nguyên không âm bj (bj ≤ 104), j = 1, 2, ..., n.

**Kết quả:** ghi ra file **LNACS.OUT**:

Ghi ra trên một dòng duy nhất độ dài của dãy con chung không liền kề dài nhất của hai dãy A và B.

Ví dụ

|  |  |
| --- | --- |
| **LNACS.INP** | **LNACS.OUT** |
| 4 5  4  9  2  4  1  9  7  3  4 | 2 |

# Palindrome

Palindrome lµ mét x©u ®èi xøng, tøc lµ mét x©u mµ ®äc tõ tr¸i sang ph¶i còng gièng nh­ ®äc tõ ph¶i sang tr¸i. B¹n cÇn viÕt mét ch­¬ng tr×nh víi mét x©u cho tr­íc, x¸c ®Þnh sè Ýt nhÊt c¸c ký tù cÇn chÌn vµo x©u ®Ó nhËn ®­îc mét Palindrome. VÝ dô, b»ng c¸ch chÌn hai ký tù vµo x©u “Ab3bd” ta nhËn ®­îc mét Palindrome (ch¼ng h¹n “dAb3bAd” hoÆc “Adb3bdA”). Tuy nhiªn, nÕu chÌn Ýt h¬n 2 ký tù th× kh«ng thÓ t¹o ®­îc Palindrome.

**D÷ liÖu vµo:** Tªn file d÷ liÖu vµo lµ PALIN.INP. Dßng thø nhÊt gåm mét sè nguyªn lµ ®é dµi N cña x©u, 3≤N≤5000. Dßng thø hai gåm mét x©u cã ®é dµi N. X©u gåm c¸c ký tù lµ c¸c ch÷ c¸i hoa A..Z, c¸c ch÷ c¸i th­êng a..z vµ c¸c ch÷ sè thËp ph©n 0..9, c¸c ch÷ c¸i hoa vµ th­êng xem nh­ lµ kh¸c nhau.

**D÷ liÖu ra:** Tªn tÖp d÷ liÖu ra lµ PALIN.OUT gåm mét sè nguyªn lµ sè l­îng ký tù tèi thiÓu cÇn chÌn vµo.

**VÝ dô:**

|  |  |
| --- | --- |
| PALIN.INP | PALIN.OUT |
| 5  Ab3bd | 2 |

**Phân đoạn tên file: CLUPART.CPP**

Cho dãy số nguyên dương tăng ngặt ***a1, a2, …, an***. Ta gọi một cách phân đoạn dãy số đã cho là việc chia dãy số đó thành các dãy con:

………………….

trong đó ***1 ≤ i1 < i2 < …. < ik-1 ≤ n***. Mỗi dãy con ***Cj*** được gọi là một đoạn. Ta gọi mật độ của ***Cj*** là số được tính theo công thức:

trong đó |***Cj***| là số lượng phần tử có trong dãy ***Cj***, còn ***max(Cj)***, ***min(Cj)*** lần lượt là số lớn nhất và nhỏ nhất trong dãy các phần tử của ***Cj***.

***Yêu cầu***: Cho số thực dương ***e***, hãy tìm cách phân đoạn dãy số đã cho thành một số ít nhất các đoạn, mỗi đoạn có một mật độ không nhỏ hơn ***e***.

INPUT: CLUPART.INP

* Dòng đầu tiên chứa số nguyên n (1 ≤ ***n*** ≤ 1000) và số thực ***e*** (***e*** < 1)
* Dòng thứ hai chứa n số nguyên ***a1, a2, …, an***, (0 < ***ai*** < 105)

OUTPUT: CLUPART.OUT

* Dòng dầu tiên chứa số ***p*** – số lượng các phân đoạn ít nhất tìm được
* Dòng thứ hai chứa ***p - 1*** số ***i1, i2, …, ip-1*** lần lượt là chỉ số phần tử cuối cùng của các đoạn 1, 2, …, ***p***-1 tính từ đầu đến cuối trong cách phân đoạn tìm được.

Ví dụ:

|  |  |
| --- | --- |
| CLUPART.INP | CLUPART.OUT |
| 5 0.36  1 7 8 12 17 | 2  1 |

**Vun sỏi Tên file: STONES.CPP**

Có N viên sỏi xếp thành 1 hàng, đống thứ i có ai viên sỏi. Ta có thể ghép hai đống kề nhau thành 1 đống và mất chi phí bằng tổng hai đống sỏi đó.

Hãy tìm cách ghép N đống sỏi này thàng một đống với chi phí nhỏ nhất.

Ví dụ: có 5 đống sỏi lần lượt là (4, 1, 2, 7, 5)

|  |  |
| --- | --- |
| 12  19  Cách ghép 1. chi phí: 41 | 19  12  Cách ghép 2. chi phí: 43 |

INPUT: STONES.INP

- Dòng 1 chứa số nguyên dương ***n*** (1 <= ***n*** <= 100)

- Dòng 2 chứa n số nguyên dương ***a1, a2, …, an*** (1 <= ***ai*** <= 1000) – là số lượng viên sỏi trong mỗi đống.

OUTPUT: STONES.OUT

- Một dòng duy nhất chứa một số nguyên dương là chi phí nhỏ nhất.

Ví dụ

|  |  |
| --- | --- |
| STONES.INP | STONES.OUT |
| 5  4 1 2 7 5 | 41 |

**Xếp hàng mua vé**

Có N người xếp hàng mua vé. Ta đánh số họ từ 1 đến N theo thứ tự đứng trong hàng. Thời gian phục vụ bán vé cho người thứ i là ti. Mỗi người cần mua một vé nhưng được quyền mua tối đa 2 vé, vì thế một số người có thể nhờ người đứng ngay trước mình mua hộ. Người thứ i nhận mua hộ vé cho người thứ i+1 thì thời gian mua vé cho 2 người là ri. Tìm phương án sao cho N người đều có vé với thời gian ít nhất.

**Dữ liệu** vào từ file văn bản TICK.INP

* Dòng thứ nhất ghi số nguyên dương N (1<N≤2000)
* Dòng thứ hai ghi N số nguyên dương t1, t2, ..., tn
* Dòng thứ ba ghi N-1 số r1, r2, ..., rn-1

**Kết quả** ghi ra file TICK.OUT

* Dòng thứ nhất ghi tổng thời gian phục vụ bán vé
* Dòng tiếp theo ghi chỉ số của các khách hàng cần rời khỏi hàng (nếu không có ai cần phải rời khỏi hàng thì qui ước ghi số 0)

# Ví dụ

|  |  |
| --- | --- |
| TICK.INP | TICK.OUT |
| 5  2 5 7 8 4  3 9 10 10 | 17  2 4 |