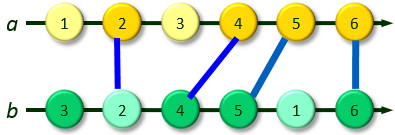
# Bài 1.Nối điểm: Tên file: NOIDIEM.CPP

Trên hai đường thẳng song song ***a*** và ***b***, người ta đánh dấu trên mỗi đường ***N*** điểm. Các điểm trên đường thẳng ***a*** được đánh số 1, 2,....,N từ trái qua phải, còn các điểm trên đường ***b*** được đánh số bởi D[1], D[2],......,D[n] là một hoán vị của N, cũng được đánh dấu từ trái qua phải (hình vẽ dưới đây cho một ví dụ khi N = 6)



Cho phép nối hai điểm thứ ***i*** trên ***a*** với điểm thứ ***j*** trên ***b*** nếu ***i*** = D[***j***].

Tìm cách nối được nhiều cặp điểm nhất với điều kiện các đoạn nối không được cắt nhau.

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

* Dòng đầu tiên chứa số nguyên N (N<=10000)
* Dòng thứ hai chứa các số D[1], D[2],......,D[n]

**Kết quả :** Ghi file văn bản NOIDIEM.OUT :

* Dòng đầu tiên chứa K là số lượng cặp điểm nối tìm được.
* K dòng tiếp theo, mỗi dòng một cặp số là số thứ tự của ***a***, số thứ tự của ***b*** (thứ tự các cặp được liệt kê theo thứ tự tăng dần của ***a***).

|  |  |
| --- | --- |
| NOIDIEM.INP | NOIDIEM.OUT |
| 6  3 2 4 5 1 6 | 4  2 2  4 3  5 4  6 6 |

Ví dụ:

# Bài 2. Robot thu quà: Tên file: ROBOT.CPP

Cuộc thi RobotCam là một cuộc thi lớn về robots được tổ chức hàng năm ở hành tinh XYZ. Sân chơi có thể mô tả trên mặt phẳng với hệ toạ độ Decattes vuông góc. Luật chơi được mô tả như sau: Trên mặt phẳng đặt *n* ≤ 100 phần quà tại các điểm có toạ độ nguyên. Các đội tham gia cuộc thi phải dùng Robot của mình đặt ở gốc tọa độ di chuyển để thu nhặt các phần quà. Vấn đề trở nên khó khăn hơn đối với các đội chơi là Robot không được di chuyển một cách tuỳ ý mà phải tuân thủ các điều kiện sau:

* Đường đi của mỗi robot phải bắt đầu và kết thúc tại các điểm trong số điểm đã cho.
* Trong quá trình di chuyển, mỗi robot không được đi sang điểm có hoành độ/tung độ nhỏ hơn hoặc bằng hoành độ/tung độ của điểm đang đứng

Yêu cầu: Em là một thành viên thiết kế Robot tham gia cuộc thi. Hãy tìm cách điều khiển Robot của đội mình để thu nhặt nhiều phần quà nhất.

|  |  |
| --- | --- |
| Robot.inp | Robot.out |
| 12  1 1  3 5  3 7  5 2  6 7  7 6  8 3  9 10  10 4  11 6  12 5  13 9 | 6  1 4 7 9 10 12 |

Input: đọc từ file Robot.inp trong đó:

* Dòng 1 ghi số N là số quà trên mặt phẳng.
* N dòng tiếp theo mỗi dòng ghi hai số x và y là tọa độ của các phần quà trên mặt phẳng

Output: Ghi ra file Robot.out:

* Dòng 1 ghi số quà nhiều nhất
* Dòng 2 là chỉ số của các phần quà mà robot thu nhặt được

Ví dụ:

# Bài 3. Bố trí phòng họp: Tên file: PHONGHOP.CPP

Có n cuộc họp, cuộc họp thứ i bắt đầu vào thời điểm a[i] và kết thúc ở thời điểm b[i]. Do chỉ có một phòng hội thảo nên 2 cuộc họp bất kì sẽ được cùng bố trí phục vụ nếu khoảng thời gian làm việc của chúng chỉ giao nhau tại đầu mút. Hãy bố trí phòng họp để phục vụ được nhiều cuộc họp nhất.

Dữ liệu: từ file PHONGHOP.INP.

* Dòng đầu là số n (1 ≤ N ≤ 1000) là số cuộc họp.
* N dũng tiếp theo mỗi dũng ghi 2 số ai và bi (1 ≤ i ≤ N) cách nhau một dấu trắng là thời điểm ban đầu và kết thúc cuộc họp thứ i.

Kết quả: PHONGHOP.OUT

* Dòng thứ nhất ghi số lượng lớn nhất các cuộc họp
* Dòng thứ hai ghi số hiệu của cuộc họp.

Ví dụ:

|  |  |
| --- | --- |
| PHONGHOP.INP | PHONGHOP.OUT |
| 7  1 4  2 5  1 7  6 8  4 10  5 8  12 16 | 3  1 4 7 |

# Bài 4. Thuê máy Tên file: THUEMAY.CPP

Trung tâm tính toán hiệu năng cao nhận được đơn đặt hàng của n khách hàng. Khách hàng i muốn sử dụng máy trong khoảng thời gian từ ai đến bi và trả tiền thuê là ci.

Hãy bố trí lịch thuê máy để tổng số tiền thu được là lớn nhất mà thời gian sử dụng máy của 2 khách hàng bất kì được phục vụ đều không giao nhau (cả trung tâm chỉ có một máy cho thuê).

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

* Dòng đầu là số N (0 < N ≤ 1000)
* N dòng tiếp theo dòng thứ i ghi 3 số A[i], B[i], C[i] cách nhau bởi một dấu cách.

Kết quả: ghi ra file **THUEMAY.OUT**

* Dòng đầu tiên ghi hai sô nguyên theo thứ tự là số lượng khách hàng được phục vụ và tổng tiền thu được
* Dòng thứ hai là chỉ số của các khách hàng được phục vụ

Ví dụ :

|  |  |
| --- | --- |
| THUEMAY.INP | THUEMAY.OUT |
| 4  400 821 800  200 513 500  100 325 200  600 900 600 | 2 1100  2 4 |

# Bài 5. Phân tích nguyên tố Tên file: ADDPRIME.CPP

Mỗi số nguyên lớn hơn 1 đều có thể biểu diễn duy nhất dưới dạng tích của các số nguyên tố (nếu như liệt kê các thừa số theo thứ tự không giảm). Thế nhưng nếu ta định biểu diễn các số nguyên dương dưới dạng tổng của các số nguyên tố (các số hạng được liệt kê theo thứ tự không giảm), thì có thể tồn tại nhiều cách biểu diễn khác nhau. Chẳng hạn, đối với số 11 có tất cả 6 cách biểu diễn.

**11** = 11

= 2 + 2 + 7

= 3 + 3 + 5

= 2 + 2 + 2 + 5

= 2 + 3 + 3 + 3

= 2 + 2 + 2 + 2 + 3

**Yêu cầu :** Cho trước số nguyên dương N (1<=N<=5000) hãy tính số lượng cách biểu diễn N dưới dạng tổng của các số nguyên tố.

**Dữ liệu :** Vào từ file văn bản ADDPRIME.INP chứa số nguyên dương N.

**Kết quả :** Ghi ra file văn bản ADDPRIME.OUT số lượng cách biểu diễn N dưới dạng tổng của các số nguyên tố. Do kết quả có thể rất lớn nên kết quả lấy bằng cách chia dư cho 1000000007.

**Ví dụ :**

|  |  |
| --- | --- |
| ADDPRIME.INP | ADDPRIME.OUT |
| 11 | 6 |

# Bài 6. CHIA VÀNG Tên file: DIVGOLD.\*\*\*

Bessie và Canmu tìm được một bao tải có N (1≤N≤250) đồng tiền vàng và chúng quyết định sẽ chia số tiền này thành hai phần đều nhất có thể. Đồng tiền thứ i có giá trị Vi (1≤Vi≤2000). Hai con bò sẽ cố gắng chia thành hai phần với tổng giá trị trong mỗi phần bằng nhau. Nếu như không thể làm được điều này chúng sẽ chia thành hai phần sao cho chênh lệch giữa phần lớn với phần nhỏ là nhỏ nhất có thể.

Bessie và Canmuu nhận thấy rằng có thể có nhiều cách để làm điều trên và chúng muốn đếm xem có bao nhiêu cách chia các đồng tiền thành hai phần với sự chênh lệch giá trị là nhỏ nhất. Chú ý rằng trong trường hợp không phân chia được đều nhau thì Bessie luôn được chia phần lớn hơn.

Ví dụ, giả sử có 5 đồng tiền với mệnh giá 2, 1, 8, 4 và 16. Bessie và Cnmuu sẽ chia thành hai phần: phần của Bessie có 1 đồng trị giá 16 và phần của Cnmuu là các đồng tiền còn lại với giá trị 2+1+8+4=15 và chênh lệch giữa hai phần là 16-15=1. Ngoài ra đây là cách chia duy nhất nên chỉ có 1 cách chia như vậy.

Chú ý rằng những đồng tiền có cùng giá trị sẽ làm cho số cách chia tăng lên nhanh chóng. Ví dụ như với tập hợp (1,1,1,1) sẽ có 6 cách chia thành hai phần bằng nhau.

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

* Dòng đầu tiên ghi số nguyên dương N
* N dòng tiếp theo, dòng thứ i ghi số Vi

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

* Dòng 1 ghi chênh lệch nhỏ nhất giữa phần lớn và phần nhỏ
* Dòng 2 ghi số cách chia tìm được. Giá trị này có thể rất lớn nên chỉ cần in phần dư của nó khi chia cho 106.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| DIVGOLD.INP | DIVGOLD.OUT |  |
| 5  2  1  8  4  16 | 1  1 |  |

# Bài 7. Bài toán chia kẹo Tên file: CHIAKEO.CPP

Cho n gói kẹo gói thứ i có A[i] cái kẹo.Yêu cầu hãy tìm cách chia các gói kẹo này thành hai phần sao cho độ chênh lệch giữa tổng số kẹo ở hai phần là nhỏ nhất có thể.

**Dữ liệu vào: chiakeo.inp**

* Dòng đầu là số n (1 ≤ n ≤ 100)
* Dòng thứ hai chứa n số nguyên A[i] là số kẹo có trong gói thứ i (A[i]<=200)

**kết quả: chiakeo.out**

* Dòng đầu tiên là độ chênh lệch min.
* Dòng hai chứa số hiện các gói kẹo thuộc phần thứ nhất theo thứ tự tăng dần.

Ví dụ:

|  |  |
| --- | --- |
| **Chiakeo.inp** | **Chiakeo.out** |
| 4  1 4 11 5 | 1  1 2 4 |

# Bài 8. Dãy con tổng chia hết cho k Tên file: DAYK.CPP

Cho một dãy số nguyên A1, A2,.., AN và một số k. Hãy tìm một dãy con (không nhất thiết phải liên tiếp nhau) dài nhất có tổng các số chia hết cho số k.

Dữ liệu vào trong file "**dayk.inp**" có dạng:

+ Dòng 1 gồm 2 số N và k (N<=1000; k<=50)

+ Các dòng tiếp theo chứa các số của mảng A.

Kết quả ra file "**dayk.out**" gồm một dòng ghi số phần tử lớn nhất tìm được.

Ví dụ:

|  |  |
| --- | --- |
| DAYK.INP | DAYK.OUT |
| 6 5  1 2 7 3 4 5 | 5 |

# Bài 9. Xâu đối xứng tên file: PALIN.CPP

Palindrome là một xâu đối xứng, tức là một xâu mà đọc từ trái sang phải cũng giống như đọc từ phải sang trái. Bạn cần viết một chương trình với một xâu cho trước, xác định số ít nhất các ký tự cần chèn vào xâu để nhận được một Palindrome. Ví dụ, bằng cách chèn hai ký tự vào xâu “Ab3bd” ta nhận được một Palindrome (chẳng hạn “dAb3bAd” hoặc “Adb3bdA”). Tuy nhiên, nếu chèn ít hơn 2 ký tự thì không thể tạo được Palindrome.

**Dữ liệu vào:** Tên file dữ liệu vào là PALIN.INP. Dòng thứ nhất gồm một số nguyên là độ dài N của xâu, 3≤N≤5000. Dòng thứ hai gồm một xâu có độ dài N. Xâu gồm các ký tự là các chữ cái hoa A..Z, các chữ cái thường a..z và các chữ số thập phân 0..9, các chữ cái hoa và thường xem như là khác nhau.

**Dữ liệu ra:** Tên tệp dữ liệu ra là PALIN.OUT gồm một số nguyên là số lượng ký tự tối thiểu cần chèn vào.

Ví dụ:

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

# Bài 10. Dãy con không liền kề dài nhất *Tên chương trình LNACS.CPP*

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 2 4 | 2 |

# Bài 11. Rút bài. Tên chương trình: RUTBAI.CPP

Có n lá bài (3 ≤ n ≤ 100), trên mỗi lá bài ghi 1 số nguyên dương không quá 10000 các quân bài xếp thành 1 chồng. Người ta lần lược rút các lá bài bên trong chồng bài (trừ lá trên cùng và dưới cùng), mỗi lần rút 1 quân đến khi còn 2 lá bài trên và dưới cùng. Giá rút 1 lá bài là tích 3 số ghi trên lá được rút, lá dưới lá được rút, lá trên lá được rút. Khi rút hết n - 2 lá bài, ta có tổng chi phí rút bài.

Mỗi trình tự rút khác nhau có tổng chi phí rút khác nhau.

**Yêu cầu:** Hãy tính tổng chi phí nhỏ nhất khi rút hết n – 2 là bài..

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

* Dòng 1 chứa số N
* Dòng 2 chứa N số nguyên dương ≤ 10000 là các số ghi trên lá bài.

**Kết quả:** Ghi ra file **RUTBAI.OUT**: 1 số duy nhất là chi phí nhỏ nhất khi rút được n – 2 lá bài

|  |  |
| --- | --- |
| **RUTBAI.INP** | **RUTBAI.OUT** |
| 5  1 2 3 4 5 | 38 |

*Giải thích: lần 1 rút là bài thứ 2 được giá là 1 \* 2 \* 3 = 6*

*lần 2 rút lá bài thứ 3 được giá là 1 \* 3 \* 4 = 12*

*lần 3 rút lá bài thứ 4 được giá là 1 \* 4 \* 5 = 20*

*Tổng chi phí = 6 + 12 + 20 = 38*

**Bài 1 – Nối điểm**

#include <bits/stdc++.h>

#define N 10000

#define inf int(1e9)

using namespace std;

int b[N+3], vet[N+3], dp[N+3];

vector <int> kq;

int n;

int main()

{

freopen("noidiem.inp","r",stdin);

freopen("noidiem.out","w",stdout);

cin >> n;

for (int i=1; i<=n; i++) cin >> b[i];

b[0] = -inf; b[n+1] = inf;

dp[0] = 1;

for (int i=1; i<=n+1; i++)

for (int j=0; j<i; j++)

if (b[j] < b[i] && dp[i] < dp[j] + 1)

{

dp[i] = dp[j] + 1;

vet[i] = j;

}

cout << dp[n+1] - 2 << "\n";

for (int j=n+1; j>0; j = vet[j])

kq.push\_back(vet[j]);

for (int i = kq.size() - 2; i>=0; i--)

cout << b[kq[i]] << " " << kq[i] << "\n";

return 0;

}

**Bài 1 – Nối điểm – Python**

inf = int(1e9)

fi = open("noidiem.inp","r")

fo = open("noidiem.out","w")

n = int(fi.readline())

b = [-inf] + list(map(int,fi.readline().split())) + [inf];

dp = [0] \* (n+3)

dp[0] = 1;

vet = [0] \* (n+3)

for i in range(1,n+2):

for j in range(0,i):

if (b[j] < b[i] and dp[i] < dp[j] + 1):

dp[i] = dp[j] + 1;

vet[i] = j

fo.write(str(dp[n+1] - 2) + "\n")

kq = [];

j = n + 1;

while (j > 0):

kq.append(vet[j]);

j = vet[j];

for i in range(len(kq) - 2, -1, -1):

fo.write(str(b[kq[i]]) + " " + str(kq[i]) + "\n")

fi.close()

fo.close()

**Bài 2 – Robot thu quà**

#include <bits/stdc++.h>

#define N 100

#define inf int(1e9)

using namespace std;

struct diem

{

int x, y, id;

diem (){}

diem (int \_x, int \_y, int \_id)

{

x = \_x; y = \_y; id = \_id;

}

bool operator < (const diem &X) const

{

return (x < X.x) || (x == X.x && y > X.y);

}

};

diem a[N+2];

int dp[N+2], vet[N+2];

vector <int> kq;

int n;

int main()

{

freopen("robot.inp","r",stdin);

freopen("robot.out","w",stdout);

cin >> n;

for (int i=1; i<=n; i++)

{

cin >> a[i].x >> a[i].y;

a[i].id = i;

}

sort(a+1,a+n+1);

a[0] = diem(-inf,-inf,0);

a[n+1] = diem(inf,inf,n+1);

dp[0] = 1;

for (int i=1; i<=n+1; i++)

for (int j=0; j<i; j++)

if (a[i].x > a[j].x && a[i].y > a[j].y &&

dp[i] < dp[j] + 1)

{

dp[i] = dp[j] + 1;

vet[i] = j;

}

cout << dp[n+1] - 2 << "\n";

for (int j=n+1;j > 0; j = vet[j])

kq.push\_back(vet[j]);

for (int i=kq.size() - 2; i>=0; i--)

cout << a[kq[i]].id << " ";

return 0;

}

**Bài 2 – Robot thu quà – Python**

inf = int(1e9)

def cmp(x):

return (x[0],-x[1]);

fi = open("robot.inp","r")

fo = open("robot.out","w")

n = int(fi.readline())

a = [(-inf,-inf,0)]

for i in range(1,n+1):

x, y = map(int,fi.readline().split())

a.append((x,y,i))

a.append((inf,inf,n+1))

a.sort(key = cmp)

dp = [0] \* (n+3);

vet = [0] \* (n+3);

dp[0] = 1

for i in range(1,n+2):

for j in range(0,i):

if (a[j][0] < a[i][0] and a[j][1] < a[i][1] and

dp[i] < dp[j] + 1):

dp[i] = dp[j] + 1;

vet[i] = j

fo.write(str(dp[n+1] - 2) + "\n")

kq = []

j = n + 1

while (j > 0):

kq.append(vet[j]);

j = vet[j]

for i in range(len(kq) - 2, -1, -1):

fo.write(str(a[kq[i]][2]) + " ");

fi.close()

fo.close()

**Bài 6 – Chia vàng**

#include <bits/stdc++.h>

#define N 250

#define modul int(1e6)

using namespace std;

int a[N+3],s,sum,n,k;

int dp[modul+3];

int main()

{ freopen("divgold.inp","r",stdin);

freopen("divgold.out","w",stdout);

cin>>n;

for(int i=1;i<=n;i++){

cin>>a[i];

s +=a[i];

}

sum = s;

s /=2;

dp[0]=1;

for(int i=1;i<=n;i++)

for(int j=s;j>=a[i];j--)

if(dp[j-a[i]]>0)

dp[j]=(dp[j]+dp[j-a[i]])% modul;

for(int j=s;j>0;j--)

if(dp[j]>0)

{

k=j;

break;

}

int j=k;

cout << sum-2\*j << "\n";

cout<<dp[j];

return 0;

}

**Bài 6 – Chia vàng – Python (Thầy Cao Chánh Thu)**

modul=int(1e6)

fi = open("divgold.inp","r")

fo = open("divgold.out","w")

n = int(fi.readline())

a = [0]

s=0

x=0

for i in range(1,n+1):

x= int(fi.readline())

a.append(x)

s+=a[i]

sum=s

s = int(s/2)

dp = [0]\*(modul+3)

dp[0] = 1

for i in range(1,n+1):

for j in range(s,a[i]-1,-1):

if(dp[j-a[i]]>0):

dp[j]=(dp[j]+dp[j-a[i]])%modul

k=0

for j in range(s,0,-1):

if(dp[j]>0):

k=j

break

j=k

fo.write(str(sum-2\*j)+"\n")

fo.write(str(dp[j]))

fi.close()

fo.close()

**Bài 5 – Nối điểm (Thầy Cao Chánh Thu)**

#include <bits/stdc++.h>

#define N int(5e4)

#define modul int(1e9+7)

using namespace std;

int p[N+3], a[N+3];

int dp[N+3];

int n,res;

void sangngto(int n)

{

fill(p,p+n+1,1);

p[0] = p[1] = 0;

for (int i=2; i\*i <= n; i++)

if (p[i] == 1) ///neu i la so nguyen to

for (int j=i\*i; j<=n; j += i)

p[j] = 0;

for (int i=2; i<=n; i++)

if (p[i] == 1) a[++res] = i;

}

int main()

{ freopen("ADDPRIME.INP","r",stdin);

freopen("ADDPRIME.OUT","w",stdout);

cin >> n;

sangngto(n);

dp[0]= 1;

for(int i=1;i<=res;i++)

for(int j=a[i];j<=n;j++)

if(dp[j-a[i]]>0)

dp[j]=(dp[j]+dp[j-a[i]])% modul;

cout<<dp[n];

return 0;

}