

Bài 1.

Cho chuỗi ký tự chỉ gồm các ký tự Latinh in hoa và in thường ('A', ..., 'Z' và 'a', ..., 'z'). Hãy tìm cách thêm ít ký tự nhất vào cuối chuỗi để được chuỗi đối xứng.

Input

Một dòng duy nhất chứa chuỗi

Output

Chuỗi ký tự đối xứng sau khi đã thêm ít nhất các ký tự vào cuối chuỗi

Test case1:

Input: xyz; Output: xyzyx

Test case2:

Input: abac; Output: abacaba

Test case3:

Input: Race; Output: RacecaR

Test case4:

Input: aA; Output: aAa

Test case5:

Input: Aaa; Output: AaaA

Test case6:

Input: Madam; Output: MadamadaM

Test case7:

Input: AaBb; Output: AaBbBaA

Test case8:

Input: level; Output: level

Test case9:

Input: AbCd; Output: AbCdCbA

Test case10:

Input: aAa; Output: aAa

Bài 2

Cho một danh sách các phần tử gồm các số nguyên cách nhau dấu cách. Hãy viết chương trình loại bỏ phần tử trùng của danh sách, giữ nguyên thứ tự của danh sách.

Input

Danh sách các số nguyên cách nhau dấu cách

Output

Danh sách đã loại bỏ các phần tử trùng, giữ nguyên thứ tự của danh sách

Test case1:

Input: 1 2 2 3 1; Output: 1 2 3

Test case2:

Input: 5 4 3 2 1; Output: 5 4 3 2 1

Test case3:

Input: 1 2 2 3 3 3; Output: 1 2 3

Test case4:

Input: 0 0 0; Output: 0

Test case5:

Input: 10; Output: 10

Test case6:

Input: -1 -1 -2 -2 -3; Output: -1 -2 -3

Test case7:

Input: 9 8 9 8 7 6; Output: 9 8 7 6

Test case18

Input: 10 10 8 5 5 1 1; Output: 10 8 5 1

Test case9:

Input: 4 4 4 4; Output: 4

Bài 3

Viết chương trình nhập vào một ma trận kích thước $n \times m$. Viết chương trình đếm số phần tử âm của ma trận.

Input:

Dòng thứ nhất nhập số hàng ma trận: n

Dòng thứ hai nhập số cột của ma trận: m

Dòng tiếp theo nhập các phần tử của ma trận theo từng hàng, cách nhau dấu cách

Output:

Số lượng phần tử âm của ma trận

Constraints:

n, m là các số nguyên dương, các phần tử của ma trận là kiểu nguyên

	Testcase1	Testcase2	Testcase3	Testcase4	Testcase5
Input	2 2 5 6 -1 10	3 2 4 5 5 -10 5 -3	2 2 0 0 0 0	3 1 -1 3 1	2 3 -1 -4 -5 4 5 6
Output	1	2	0	1	3
	Testcase6	Testcase7	Testcase8	Testcase9	Testcase10
Input	1 1 5	2 2 3 4 5 6	4 3 4 5 7 5 -5 -1 4 7 8 100 5 6	3 3 5 6 -2 7 7 9 4 2 6	4 2 4 7 5 8 -1 -2 5 7
Output	0	0	2	1	2

Bài 4

Hãy viết chương trình nhập vào một ma trận vuông kích thước A kích thước (nxn). Ma trận được gọi là ma trận đối xứng qua đường chéo chính nếu với mọi i, j: $A[i][j]=A[j][i]$. Hãy kiểm tra ma trận có đối xứng hay không

Input:

Dữ liệu nhập vào trên n+1 dòng

Dòng 1: Nhập số nguyên n

n dòng tiếp theo: các dòng của ma trận, mỗi phần tử của ma trận được nhập theo dòng, mỗi phần tử $A[i][j]$ được cách nhau dấu cách

Output

In ra Yes nếu ma trận đối xứng

In ra No nếu ma trận không đối xứng

Constraints

- $n, A[i][j]$: số nguyên
- $1 \leq n \leq 100$

	Testcase1	Testcase2	Testcase3	Testcase4	Testcase5
Input	2 1 2 2 1	3 1 0 0 0 1 0 0 0 1	2 1 2 3 4	1 4	3 3 1 2 3 2 4 5 3 5 6
Output	Yes	Yes	No	Yes	Yes
	Testcase6	Testcase7	Testcase8	Testcase9	Testcase10
Input	3 1 2 3 4 5 6 7 8 9	2 0 0 0 0	3 1 1 0 1 1 0 0 0 1	2 -1 2 2 -1	3 1 2 4 -1 4 -2 4 5 6
Output	No	Yes	Yes	Yes	No

Bài 5

Cho một danh sách gồm n số nguyên. Hãy tách danh sách thành 2 dòng:

Dòng 1: các số chẵn theo đúng thứ tự xuất hiện

Dòng 2: các số lẻ theo đúng thứ tự xuất hiện

Nếu một nhóm không có phần tử nào, in ra dòng trống.

Input:

- Dòng thứ nhất nhập số phần tử của danh sách: n
- Dòng thứ hai nhập n số nguyên, cách nhau bởi dấu cách

Output:

- Dòng 1: các số chẵn (cách nhau bởi dấu cách)
- Dòng 2: các số lẻ (cách nhau bởi dấu cách)

Constraints:

- n là số nguyên dương
- Các phần tử là số nguyên

	Testcase1	Testcase2	Testcase3	Testcase4	Testcase5
Input	5 1 2 3 4 5	4 2 4 6 8	4 1 3 5 7	1 10	1 7
Output	2 4 1 3 5	2 4 6 8	1 3 5 7	10	7
	Testcase6	Testcase7	Testcase8	Testcase9	Testcase10
Input	6 0 -1 -2 3 4 -5	5 -2 -4 -6 -8 -10	6 -1 -3 -5 -7 -9 -11	8 1 2 2 3 3 4 4 5	10 9 8 7 6 5 4 3 2 1 0
Output	0 -2 4 -1 3 -5	-2 -4 -6 -8 -10	-1 -3 -5 -7 -9 -11	2 2 4 4 1 3 3 5	8 6 4 2 0 9 7 5 3 1

Bài 6

Viết chương trình nhập vào danh sách gồm n nhân viên. Mỗi nhân viên có các thông tin: **tên, mã, hệ số lương, phụ cấp**. Tính **lương tháng** của từng nhân viên theo công thức: $\text{LuongThang} = \text{HeSoLuong} \times 2000000$. Hãy in ra các nhân viên có **lương tháng** $> X$ theo **đúng thứ tự nhập vào**.

Input:

- Dòng thứ nhất nhập số nhân viên: n
- n dòng tiếp theo, mỗi dòng nhập thông tin 1 nhân viên theo thứ tự:
TenNV MaNV HeSoLuong PhuCap
- Dòng cuối nhập số nguyên: X (mức lương cần so sánh)

Output:

- Mỗi dòng in ra thông tin của một nhân viên thỏa điều kiện $\text{LuongThang} > X$, theo thứ tự:
MaNV TenNV HeSoLuong PhuCap LuongThang
- Nếu không có nhân viên nào thỏa điều kiện, **không in gì thêm**.

Constrains:

- $1 \leq n \leq 200$
- MaNV, PhuCap, X là số nguyên dương
- HeSoLuong là số thực dương
- TenNV không chứa khoảng trắng

	Testcase1	Testcase2	Testcase3
Input	3 An NV01 2.5 500000 Binh NV02 3.0 1000000 Cuong NV03 1.8 300000 6000000	2 Hoa NV01 2.0 0 Lan NV02 2.5 500000 4000000	3 Nam NV01 1.5 200000 Minh NV02 1.8 300000 Tuan NV03 2.0 0 5000000
Output	NV02 Binh 3.0 1000000 7000000	NV02 Lan 2.5 500000 5500000	
	Testcase4	Testcase5	Testcase5
Input	1 An NV01 3.0 1000000 5000000	4 A NV01 2.0 1000000 B NV02 2.0 500000 C NV03 2.0 0 D NV04 1.9 200000 4500000	3 Lan NV01 2.5 0 Mai NV02 2.5 500000 Ha NV03 2.5 1000000 5500000
Output	NV01 An 3.0 1000000 7000000	NV01 A 2.0 1000000 5000000 NV02 B 2.0 500000 4500000	NV02 Mai 2.5 500000 5500000 NV03 Ha 2.5 1000000 6000000
	Testcase 6	Testcase 7	Testcase 8
Input	5 A NV01 1.0 0 B NV02 1.5 200000 C NV03 2.0 300000 D NV04 2.5 400000 E NV05 3.0 500000 4000000	3 An NV01 2.0 0 Binh NV02 2.0 1000000 Cuong NV03 2.0 2000000 5000000	2 Nam NV01 2.999 1000 Bac NV02 3.0 0 6000000
Output	NV04 D 2.5 400000 5400000 NV05 E 3.0 500000 6500000	NV03 Cuong 2.0 2000000 6000000	NV02 Bac 3.0 0 6000000

Bài 7

Viết chương trình nhập vào danh sách gồm n học sinh. Mỗi học sinh có các thông tin: **tên, mã, điểm trung bình**. Một học sinh được xếp loại **Giỏi** nếu $DiemTrungBinh \geq 8.0$. Hãy đếm và in ra **số lượng học sinh loại Giỏi**.

Input:

- Dòng thứ nhất nhập số học sinh: n
- n dòng tiếp theo, mỗi dòng nhập thông tin 1 học sinh theo thứ tự:
TenHS MaHS DiemTrungBinh

Output:

- In ra một số nguyên là **số lượng học sinh có DiemTrungBinh ≥ 8.0**

Constraints:

- $1 \leq n \leq 200$

- MaHS là số nguyên dương
- DiemTrungBinh là số thực trong khoảng [0, 10]
- TenHS không chứa khoảng trắng

	Testcase1	Testcase2	Testcase3
Input	3 An HS01 7.5 Binh HS02 8.0 Cuong HS03 9.2	5 Lan HS01 8.5 Mai HS02 8.2 Hoa HS03 9.0 Nga HS04 7.9 Anh HS05 6.5	4 A HS01 5.0 B HS02 6.0 C HS03 7.5 D HS04 7.9
Output	2	3	0
	Testcase4	Testcase5	Testcase6
Input	1 An HS01 8.0	1 Binh HS01 7.99	6 A HS01 8.0 B HS02 8.0 C HS03 8.0 D HS04 8.0 E HS05 8.0 F HS06 8.0
Output	1	0	6
	Testcase 7	Testcase 8	Testcase 9
Input	5 Lan HS01 9.5 Mai HS02 9.0 Hoa HS03 8.5 Nga HS04 8.0 Anh HS05 7.0	7 A HS01 7.0 B HS02 7.5 C HS03 8.1 D HS04 8.2 E HS05 8.3 F HS06 6.9 G HS07 5.5	10 HS1 HS01 8.0 HS2 HS02 8.1 HS3 HS03 7.9 HS4 HS04 9.0 HS5 HS05 6.0 HS6 HS06 10.0 HS7 HS07 8.0 HS8 HS08 7.0 HS9 HS09 8.5 HS10 HS10 5.5
Output	4	3	5

Bài 8

Nhập n nhân viên, mỗi nhân viên gồm họ tên, hệ số lương, thâm niên công tác. Thông tin của mỗi nhân viên nhập trên một dòng, cách nhau dấu cách. Thêm lương cho mỗi nhân viên, lương = hệ số lương x 2000000 + thâm niên. Hãy in ra nhân viên có lương thấp nhất, bao gồm họ tên, hệ số lương, thâm niên và lương trên một dòng (Họ tên gồm 2 phần: hodem ten)

Input:

- Dòng thứ nhất nhập số nhân viên: n

- n dòng tiếp theo, mỗi dòng nhập thông tin 1 nhân viên theo thứ tự (cách nhau bởi dấu cách):

HoDem Ten HeSoLuong ThamNien

Trong đó **họ tên gồm 2 phần**: HoDem và Ten.

Output:

- In ra **1 dòng** thông tin của nhân viên có lương thấp nhất theo thứ tự:
HoDem Ten HeSoLuong ThamNien Luong

(Các thông tin cách nhau bởi dấu cách)

Constraints:

- $1 \leq n \leq 200$
- HeSoLuong là số thực dương
- ThamNien là số nguyên không âm
- HoDem, Ten không chứa khoảng trắng

	Testcase1	Testcase2	Testcase3
Input	3 Nguyen An 2.5 500000 Tran Binh 3.0 100000 Le Cuong 2.0 300000	2 Pham Hoa 2.0 0 Do Lan 2.0 500000	1 Nguyen Minh 3.0 200000
Output	Le Cuong 2.0 300000 4300000	Pham Hoa 2.0 0 4000000	Nguyen Minh 3.0 200000 6200000
	Testcase4	Testcase5	Testcase6
Input	3 Hoang Long 2.1 200000 Vu Tuan 2.0 300000 Pham Kien 2.2 0	3 Le Anh 2.999 1000 Tran Binh 3.0 0 Pham Cuong 2.5 500000	4 Nguyen Nam 1.8 200000 Do Son 1.9 0 Tran Hung 1.8 100000 Le Tien 2.0 0
Output	Vu Tuan 2.0 300000 4300000	Le Anh 2.999 1000 5999000	Tran Hung 1.8 100000 3700000
	Testcase 7	Testcase 8	Testcase 9
Input	4 Nguyen Ha 1.5 0 Tran My 1.4 100000 Le Phuong 1.4 100000 Pham Quang 2.0 0	5 Nguyen Van 2.0 100000 Tran Thi 1.8 300000 Le Minh 1.8 300000 Pham An 2.5 0 Do Khoa 3.0 0	3 Hoang Nam 1.0 500000 Vu Anh 1.0 500000 Nguyen Long 1.2 0
Output	Tran My 1.4 100000 2900000	Tran Thi 1.8 300000 3900000	Hoang Nam 1.0 500000 2500000

Bài 9

Để chuẩn bị cho lễ giáng sinh sắp đến, bạn Nam định trang trí cho nhà của mình một cây thông Noel có 3 tầng.

Tầng thứ nhất là các lá thông tạo thành hình tam giác cân có chiều cao là n . Dọc theo 2 cạnh của cây thông, Nam trang trí thêm các đèn LED nhấp nháy. Các lá thông biểu diễn bằng ký tự 'x', đèn LED biểu diễn bằng ký tự '#'. Tính cả đèn và lá thông thì tại mỗi dòng sẽ có số lượng lần lượt là 1, 3, 5, ..., $2*n-1$.

Tầng thứ hai và tầng thứ ba cũng tương tự như tầng thứ nhất nhưng lớn hơn. Tầng thứ hai có $n+1$ dòng và tầng thứ 3 có $n+2$ dòng.

Input:

3

Output

```
#
#x#
#xxx#
#
#x#
#xxx#
#xxxxx#
#
#x#
#xxx#
#xxxxx#
#xxxxxxxx#
```

	Testcase1	Testcase2	Testcase3
Input	1	2	3
Output	<pre># # #x# # #x# #xxx#</pre>	<pre># #x# # #x# #xxx# # #x# #xxx# #xxxxx#</pre>	<pre># #x# #xxx# # #x# #xxx# #xxxxx# # #x#</pre>

			#xxx# #xxxxx# #xxxxxxxx#
	Testcase4	Testcase5	Testcase6
Input	4	5	6
Output	# #x# #xxx# #xxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxxxx#	# #x# #xxx# #xxxxx# #xxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxx# #xxxxxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx#	# #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# # #x# #xxx# #xxxxx# #xxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx# #xxxxxxxxxxx#

Bài 10

Viết chương trình nhập vào danh sách gồm n đơn hàng.

Mỗi đơn hàng gồm các thông tin: **mã đơn hàng, tên khách hàng, số lượng, đơn giá.**

Tính **doanh thu** của mỗi đơn hàng theo công thức:

$\text{DoanhThu} = \text{SoLuong} \times \text{DonGia}$

$\text{DoanhThu} = \text{SoLuong} \times \text{DonGia}$

Thực hiện các yêu cầu sau:

1. **Loại bỏ** các đơn hàng có $\text{SoLuong} = 0$
2. **Sắp xếp** danh sách đơn hàng còn lại theo **doanh thu giảm dần**
3. Nếu nhiều đơn hàng có cùng doanh thu, sắp xếp tiếp theo **mã đơn hàng tăng dần**
4. In ra danh sách kết quả

Input

- Dòng thứ nhất nhập số nguyên n – số lượng đơn hàng
- n dòng tiếp theo, mỗi dòng nhập thông tin của một đơn hàng theo thứ tự: **mã đơn hàng, tên khách hàng, số lượng, đơn giá.**

Output:

- Dòng thứ nhất in ra: Danh sách đơn hàng hợp lệ: (k đơn hàng, trong đó k là số đơn hàng còn lại sau khi loại bỏ)
- k dòng tiếp theo, mỗi dòng in ra thông tin của một đơn hàng theo thứ tự đã sắp xếp, theo định dạng: MaDonHang TenKhachHang SoLuong DonGia DoanhThu

Constraints:

- $1 \leq n \leq 200$
- MaDonHang là số nguyên dương, không trùng nhau
- TenKhachHang không chứa khoảng trắng
- SoLuong, DonGia là số nguyên không âm
- Doanh thu vừa trong kiểu số nguyên

	Testcase1	Testcase2	Testcase3
Input	4 101 An 5 10000 102 Binh 0 20000 103 Cuong 3 15000 104 Dung 5 10000	3 1 A 0 1000 2 B 0 2000 3 C 0 3000	5 10 Nam 2 50000 11 Hoa 4 25000 12 Lan 1 100000 13 Minh 2 50000 14 An 3 10000
Output	Danh sách đơn hàng hợp lệ: 3 101 An 5 10000 50000 104 Dung 5 10000 50000 103 Cuong 3 15000 45000	Danh sách đơn hàng hợp lệ: 0	Danh sách đơn hàng hợp lệ: 5 10 Nam 2 50000 100000 11 Hoa 4 25000 100000 12 Lan 1 100000 100000 13 Minh 2 50000 100000 14 An 3 10000 30000
	Testcase4	Testcase5	Testcase6
Input	1 99 Test 10 1000	6 1 A 5 20000 2 B 3 30000 3 C 10 10000 4 D 0 50000 5 E 1 60000 6 F 2 30000	4 20 Khoa 1 99999 21 Linh 2 50000 22 Trang 3 33333 23 Mai 0 100000
Output	Danh sách đơn hàng hợp lệ: 1 99 Test 10 1000 10000	Danh sách đơn hàng hợp lệ: 5 3 C 10 10000 100000 2 B 3 30000 90000 6 F 2 30000 60000 5 E 1 60000 60000	Danh sách đơn hàng hợp lệ: 3 21 Linh 2 50000 100000 22 Trang 3 33333 99999 20 Khoa 1 99999 99999

		1 A 5 20000 100000	
	Testcase 7		
Input	3 100 An 1 100000 101 Binh 2 50000 102 Cuong 4 25000		
Output	Danh sach don hang hop le: 3 100 An 1 100000 100000 101 Binh 2 50000 100000 102 Cuong 4 25000 100000		

Bài 11

Để tính điểm rèn luyện cho sinh viên cuối học kì nhà trường quy định: Sinh viên phải tham gia các hoạt động để tích lũy điểm, mỗi hoạt động có số điểm tích lũy được quy định trước. Để có hạnh kiểm tốt, sinh viên phải có tổng điểm rèn luyện ≥ 70 điểm.

Viết chương trình nhập vào danh sách sinh viên và các hoạt động mà sinh viên đã tham gia. In ra danh sách các sinh viên chưa đủ điểm rèn luyện để đạt hạnh kiểm tốt cuối kì nhằm cảnh báo các bạn tích cực tham gia các hoạt động trong nhà trường.

Input:

+ Dòng đầu tiên nhập số nguyên dương n là số bộ dữ liệu về hoạt động của sinh viên ($1 \leq n \leq 200$).

+ n dòng tiếp theo, mỗi dòng nhập một bộ thông tin gồm: mã sinh viên, tên sinh viên, mã hoạt động. Các thông tin nhập vào đều kiểu chuỗi, không chứa dấu cách và cách nhau một khoảng trắng.

Chú ý: Các sinh viên có thể tham gia một mã hoạt động nhiều lần (ở các thời điểm khác nhau).

Output: Danh sách các sinh viên có **tổng điểm rèn luyện đã tích lũy < 70** theo thứ tự: **mã sinh viên, tên sinh viên, tổng điểm rèn luyện**. Mỗi sinh viên trên một dòng. Nếu không có sinh viên nào không đủ điểm đạt hạnh kiểm tốt, in ra thông báo "**Khong co sinh vien nao diem ren luyen < 70** ".

Biết, danh sách các hoạt động và điểm rèn luyện tích lũy trong kỳ cho trong bảng sau:

Mã HĐ	Điểm rèn luyện
-------	-------------------

hd1	5
-----	---

hd2 8
hd3 10
hd4 10
hd5 15

For example:

Test	Input	Result
2	6 SV01 An hd5 SV01 An hd5 SV01 An hd5 SV01 An hd5 SV01 An hd5 SV01 An hd5	Khong co sinh vien nao diem ren luyen < 70

Bài 12

Viết chương trình nhập vào danh sách gồm n hình tròn trong mặt phẳng. Nhập vào một điểm A (x_a, y_a) và điểm B (x_b, y_b). Cho biết điểm A, B nằm trong, nằm trên, hay nằm ngoài các hình tròn đã nhập vào. Biết thông tin của hình tròn gồm mã hình tròn, bán kính, tọa độ tâm (x, y).

Input:

- + Dòng thứ nhất nhập vào số nguyên dương n .
- + Dòng tiếp theo nhập vào thông tin của từng hình tròn. Thứ tự nhập vào là mã, tâm x , tâm y , bán kính.
- + Dòng cuối cùng nhập vào x_a, y_a, x_b, y_b cách nhau dấu cách.

Output:

- + Gồm $2n$ dòng, mỗi dòng in ra thông báo:

"Diem (x_a, y_a) nam trong/tren/ngoai hinh tron tam (x, y) ban kinh r "

"Diem (x_b, y_b) nam trong/tren/ngoai hinh tron tam (x, y) ban kinh r "

+ Trong đó: (x_a, y_a) và (x_b, y_b) là tọa độ điểm A và B, (x, y) và r là tọa độ tâm và bán kính các hình tròn trong danh sách.

Constraints:

+ $1 \leq n \leq 100$.

+ bán kính là số thực với độ chính xác 3 chữ số thập phân, tọa độ là các số nguyên.

+ Số pi = 3.14159.

For example:

Test	Input	Result
1	5 88 17 57 9.000 32 69 54 11.000 21 76 29 30.000 68 25 55 31.000 34 41 12 16.667 45 60 18 53	<p>Điểm (45, 60) nằm ngoài hình tròn tâm (17, 57) bán kính 9.000</p> <p>Điểm (18, 53) nằm trong hình tròn tâm (17, 57) bán kính 9.000</p> <p>Điểm (45, 60) nằm ngoài hình tròn tâm (69, 54) bán kính 11.000</p> <p>Điểm (18, 53) nằm ngoài hình tròn tâm (69, 54) bán kính 11.000</p> <p>Điểm (45, 60) nằm ngoài hình tròn tâm (76, 29) bán kính 30.000</p> <p>Điểm (18, 53) nằm ngoài hình tròn tâm (76, 29) bán kính 30.000</p> <p>Điểm (45, 60) nằm trong hình tròn tâm (25, 55) bán kính 31.000</p> <p>Điểm (18, 53) nằm trong hình tròn tâm (25, 55) bán kính 31.000</p> <p>Điểm (45, 60) nằm ngoài hình tròn tâm (41, 12) bán kính 16.667</p> <p>Điểm (18, 53) nằm ngoài hình tròn tâm (41, 12) bán kính 16.667</p>

Bài 13

Cho một số nguyên n có đúng 4 chữ số, trong đó ít nhất hai chữ số khác nhau.

Yêu cầu

Thực hiện lặp lại các bước sau với số n :

1. Sắp xếp các chữ số của n theo **thứ tự giảm dần** để được số L .
2. Sắp xếp các chữ số của n theo **thứ tự tăng dần** để được số S .
3. Gán $n \leftarrow L - S$.

Quá trình được lặp lại cho đến khi **giá trị n không còn thay đổi giữa hai bước liên tiếp**.

Hãy:

- In ra **giá trị cố định cuối cùng** của n .
- In ra **toàn bộ các phép trừ** đã thực hiện theo đúng thứ tự.

Các số phải được in ở **dạng 4 chữ số**, có thể thêm chữ số 0 ở đầu nếu cần.

Input

- Một số nguyên n có đúng 4 chữ số, trong đó có ít nhất hai chữ số khác nhau.

Output

- Dòng 1: Giá trị cố định cuối cùng của n .
- Các dòng tiếp theo: Mỗi dòng in một phép trừ theo định dạng $L - S = n$

Bài 14

Một dãy được gọi là dãy sóng nếu các phần tử lần lượt tăng rồi giảm hoặc giảm rồi tăng.

Yêu cầu: Cho một dãy gồm n số nguyên. Hãy kiểm tra xem dãy đã cho có phải là dãy sóng hay không.

Input:

- Dòng 1: Số nguyên n ($1 \leq n \leq 1000$).
- Dòng 2: n số nguyên, các nhau bởi dấu cách.

Output: In ra YES nếu là dãy sóng. In ra NO nếu không phải.

Constraints: Dãy có 1 phần tử không gọi là dãy sóng.

For example:

Test	Input	Result
1	5 1 3 2 4 3	YES
3	4 1 2 3 4	NO

Bài 15

Cho một dãy gồm **n số nguyên**. Kiểm tra xem dãy số có phải là **dãy tăng chặt** hay không. Biết dãy tăng chặt là dãy có mọi phần tử đứng sau **lớn hơn** phần tử đứng ngay trước nó.

Input

- Dòng 1: Số nguyên n ($1 \leq n \leq 1000$)
- Dòng 2: n số nguyên, cách nhau bởi dấu cách.

Output

- In ra YES nếu dãy tăng chặt
- In ra NO nếu ngược lại

For example:

Test	Input	Result
1	5 1 2 3 4 5	YES
2	5	NO

Test	Input	Result
	1 1 2 3 4	

Bài 16

Cho một danh sách gồm **n số nguyên**.

Yêu cầu: Hãy tạo một **danh sách mới**, trong đó:

- Phần tử thứ **i** của danh sách mới bằng tổng các phần tử từ vị trí 1 đến vị trí **i** của danh sách ban đầu.

Input:

- Dòng thứ nhất: số nguyên **n** ($1 \leq n \leq 1000$)
- Dòng thứ hai: **n** số nguyên, mỗi số có giá trị tuyệt đối không vượt quá 10^6

Output:

- In ra **n** số nguyên của danh sách tích lũy, các phần tử cách nhau bởi dấu cách.

Bài 17

Ban đầu, Nam có **N** chiếc bánh. Sau khi ăn mỗi chiếc bánh, Nam sẽ thu được 1 vỏ bánh.

Cửa hàng cho phép:

- Với mỗi **K** vỏ bánh, Nam có thể đổi được 1 chiếc bánh mới.
- Bánh đổi được tiếp tục ăn và vỏ bánh của bánh đổi cũng được tính để đổi tiếp.

Yêu cầu: Hãy tính tổng số bánh mà Nam có thể ăn được.

Input: Một dòng chứa hai số nguyên **N** và **K**

- $1 \leq N \leq 1000$
- $2 \leq K \leq 100$

Output: In ra một số nguyên duy nhất là tổng số bánh Nam ăn được.

For example:

Test	Input	Result
3	3 2	5

Cho 2 danh sách số nguyên A và B. Gộp chúng lại rồi sắp xếp theo thứ tự tăng dần.

Bài 18

Cho hai danh sách số nguyên **A** và **B**.

Yêu cầu: Hãy gộp hai danh sách A và B thành một danh sách duy nhất, sau đó sắp xếp các phần tử theo thứ tự tăng dần.

Input:

- Dòng thứ nhất: số nguyên **n** – số phần tử của danh sách **A**
- Dòng thứ hai: **n** số nguyên của danh sách **A**
- Dòng thứ ba: số nguyên **m** – số phần tử của danh sách **B**
- Dòng thứ tư: **m** số nguyên của danh sách **B**

$$1 \leq n, m \leq 1000$$

Output: In ra danh sách sau khi gộp và sắp xếp tăng dần, các phần tử cách nhau bởi dấu cách.

For example:

Test	Input	Result
1	3 1 3 5 3 2 4 6	1 2 3 4 5 6

Bài 19

Cho một số nguyên dương **n** ($1 \leq n \leq 10^6$).

Yêu cầu

Thực hiện lặp lại các bước sau với số n :

1. Tách các chữ số của n .
2. Tính tổng **bình phương** các chữ số đó, gọi là S .
3. Gán $n \leftarrow S$.

Quá trình được lặp lại cho đến khi:

- $n = 1$ (khi đó n là **số hạnh phúc**), hoặc
- n lặp lại một giá trị đã xuất hiện trước đó (khi đó n **không hạnh phúc**).

Hãy kiểm tra và cho biết n có phải là số hạnh phúc hay không.

Input

- Một số nguyên dương n ($1 \leq n \leq 10^6$).

Output

- In ra HAPPY nếu n là số hạnh phúc.
- In ra UNHAPPY nếu n không phải là số hạnh phúc.

For example:

Test	Input	Result
2	19	HAPPY

BÀI 20

Mỗi khi mua **1 chiếc bánh** với giá **5.000 đồng**, khách hàng sẽ được **1 điểm tích lũy**.

Cứ đủ **K điểm**, khách hàng có thể đổi được **1 chiếc bánh miễn phí**.

Lưu ý:

- Bánh đổi miễn phí vẫn được tính là 1 bánh ăn được
- Bánh đổi miễn phí cũng được tặng điểm như khi mua bằng tiền

Yêu cầu:

Với số tiền ban đầu là **T đồng**, hãy tính **tổng số bánh tối đa** mà khách hàng có thể ăn được.

Input: Một dòng chứa hai số nguyên **T** và **K**

- $1 \leq T \leq 10^6$
- $2 \leq K \leq 1000$

Output: In ra một số nguyên là tổng số bánh ăn được.

For example:

Test	Input	Result
1	5000 2	1