**Bài tập về nhà Lập Trình C/C++ cơ bản < 3**

**BTCB.01: Gán kiểu dữ liệu 01**

Viết chương trình để thêm kiểu dữ liệu chính xác cho các biến dưới đây:

a = 5 ; f = -1 ;

b = 10 ; g = -12345 ;

c = 100000000 ; h = -1111111111 ;

d = 2000000000 ; i = -1111111111111 ;

e = 0 ; j = -1234567890000000 ;

**BTCB.02: Gán kiểu dữ liệu 02**

Viết chương trình để thêm kiểu dữ liệu chính xác cho các biến dưới đây:

ca = 'a' ; sa = "string" ;

cb = 'A' ; sb = "I'm Blue!" ;

cc = ' ' ; sc = " " ;

cd = '' ; sd = " " ;

ce = '\*' ; se = "Special char: \*^%^";

**BTCB.03: Gán kiểu dữ liệu 03**

Viết chương trình để thêm kiểu dữ liệu chính xác cho các biến dưới đây:

ba = 0 ; fa = 0.00002 ;

bb = 1 ; fb = 222 ;

bc = true ; fc = -123.45 ;

bd = false ; fd = -123.0000001111003 ;

be = 100 ; fe = 66.8888888888888 ;

**BTCB.04: Ép kiểu 01**

Viết chương trình nhập vào 2 số nguyên a, b. Đưa ra kết quả a/b.

Ví dụ:

a = 3, b = 2 => Kết quả: 1.5

**BTCB.05: Ép kiểu 02**

Viết chương trình nhập vào một ký tự. Đưa ra vị trí của ký tự đó trong bảng mã ASCII.

Ví dụ:

ký tự nhập là ‘a’ => kết quả in ra là: 97

**BTCB.06: Ép kiểu 03**

Viết chương trình nhập vào 2 số nguyên a, b. Đưa ra kết quả của a/b làm tròn xuống.

Ví dụ:

Nhập 3 2 => kết quả: 1

**BTCB.07: Ép kiểu 04**

Viết chương trình nhập vào 2 số nguyên a, b. Đưa ra kết quả của a/b làm tròn lên.

Ví dụ:

Nhập 3 2 => kết quả: 2

**BTCB.08: Ép kiểu 05**

Viết chương trình nhập vào 1 **ký tự** là số trong khoảng từ 0-9. Đưa ra kết quả phép cộng của nó với số 8 dưới dạng số nguyên .

Ví dụ:

Nhập: **2** => kết quả: **10**

**BTCB.09: Tính trung bình cộng của 4 số nguyên**

Viết chương trình nhập vào 4 số nguyên a,b,c,d . Đưa ra kết quả trung bình cộng của 4 số đó. ( a,b,c,d <= 1e12)

Ví dụ:

Nhập : **1, -1, 10, -10** => kết quả: **0**

**BTCB.10: Tính diện tích và chu vi hình chữ nhật**

Viết chương trình nhập vào 2 số nguyên a, b ( lần lượt là chiều dài vào chiều rộng của hình chữ nhật . ( 0 < a, b <= 1e9)

Đưa ra chu vi và diện tích của hình chữ nhật tương ứng theo định dạng .

Ví dụ:

Nhập : **Nhập chiều dài: 2**

**Nhập chiều rộng : 4**

=> Kết quả: **Chu vi hình chữ nhật đó là: 12 .**

**Diện tích hình chữ nhật đó là: 8 .**

**BTCB.11: Tính lũy thừa của 2**

Viết chương trình nhập vào 1 số nguyên n .

Đưa ra kết quả là lũy thừa của 2. ( 0 <= n < 32)

Ví dụ:

Nhập : **0** => kết quả: **1**

**BTCB.12: Chia đôi**

Viết chương trình nhập vào 1 số nguyên n .

Sử dụng thao tác với bit, đưa ra kết quả là của số n/2. ( 0 <= n < = 1e18)

Ví dụ:

Nhập : **5** => kết quả: **2**

**BTCB.13: Kiểm tra chẵn lẻ**

Viết chương trình nhập vào 1 số nguyên dương n .( 0 <= n < = 1e18)

Kiểm tra xem số nhập vào là chẵn hay lẻ theo định dạng sau.

Ví dụ:

Nhập : **So can nhap la : 5**

=> kết quả: **5 la so le .**

**BTCB.14: Tính phân số**

Viết chương trình nhập vào 2 số nguyên lần lượt x, y ( -1e9 <= x, y < = 1e9)

Đưa ra 1 số nguyên , kết quả của phân số (x+y) / (x-y) .

Ví dụ:

Nhập : **2 4**

=> kết quả: **-3**

**BTCB.15: Căn 2021**

Viết chương trình nhập 1 số nguyên a ( -1e9 <= a < = 1e9)

Đưa ra 1 giá trị là căn bậc 2021 của số đó bình phương cộng 1 .

Ví dụ:

Nhập : **Nhap so a: 6**

**=>** kết quả: **1.00179**

**BTCB.16: Giải phương trình bậc nhất**

Viết chương trình nhập 2 số nguyên a,b ( -1e9 <= a, b < = 1e9)

Đưa ra 1 giá trị là nghiệm của phương trình a\*x + b = 0 ;

Ví dụ:

Nhập : **Nhap so a : 2**

**Nhap so b : 1**

=> kết quả: **Phuong trinh co 1 nghiem la x: -0.5**

**BTCB.17: Giải phương trình bậc hai**

Viết chương trình nhập 3 số nguyên a,b,c ( -1e9 <= a, b, c < = 1e9 ; a != 0)

Đưa ra 1 giá trị là nghiệm của phương trình ax² +bx + c = 0. (a≠0)

Ví dụ:

Nhập : **Nhap a (a!=0): 3**

**Nhap b: -5**

**Nhap c: 2**

=> kết quả: **Phuong trinh da cho co hai nghiem phan biet la:**

**x1=1**

**x2=0.666667**

**BTCB.18: Max 2 số**

Viết chương trình nhập vào 2 số nguyên lần lượt a , b ( -1e9 <= a, b < = 1e9)

Đưa ra 1 số nguyên là số lớn nhất trong 2 số ấy

Ví dụ:

Nhập : **1000 1**

=> kết quả: **So lon nhat trong 2 so do la: 1000 ;**

**BTCB.19: Min 2 số**

Viết chương trình nhập vào 2 số nguyên lần lượt a, b ( -1e9 <= a, b < = 1e9)

Đưa ra 1 số nguyên là số nhỏ nhất trong 2 số ấy

Ví dụ:

Nhập : **1000 1**

=> kết quả: **So lon nhat trong 2 so do la: 1 ;**

**BTCB.20: Max 3 số**

Viết chương trình nhập vào 3 số nguyên lần lượt a, b,c ( -1e9 <= a, b, c < =1e9)

Đưa ra 1 số nguyên là số lớn nhất trong 3 số ấy

Ví dụ:

Nhập :  **6 1 10**

=> kết quả: **So lon nhat trong 3 so do la: 10**

**BTCB.21: Min 3 số**

Viết chương trình nhập vào 3 số nguyên lần lượt a, b,c ( -1e9 <= a, b, c < =1e9)

Đưa ra 1 số nguyên là số nhỏ nhất trong 3 số ấy

Ví dụ:

Nhập :  **6 1 10**

=> kết quả: **So nho nhat trong 3 so do la: 1**

**BTCB.22: Check tháng**

Viết chương trình hàm nhập vào 2 số là tháng , năm hợp lệ và cho biết tháng đó có bao nhiêu ngày?

Ví dụ:

Nhập :  **Nhap thang: 2**

**Nhap nam: 2020**

=> kết quả: **Thang 2 co 29 ngay!**

**BTCB.23: Tìm max 4 số**

Viết chương trình nhập vào 4 số nguyên a, b, c, d. Đưa ra kết quả là số có giá trị lớn nhất trong 4 số đó.

Ví dụ:

Nhập: **So a: 2**

**So b: 3**

**So c: 6**

**So d: 1**

=> Kết quả: **So max la: 6**

**BTCB.24: Bảng xếp hạng**

Viết chương trình nhập vào một số thực x là số điểm của một học sinh. Hãy cho biết học sinh đó thuộc loại nào ? Biết rằng phổ điểm như sau:

Từ 0 <= x <= 5 : Yếu

Từ 5 < x <= 6.5 : Trung bình

Từ 6.5 < x < 8 : Khá

Từ 8 <= x < 9 : Giỏi

Từ 9 <= x < 10 : Xuất sắc

Ví dụ: **Nhap diem: 7.0**

Kết quả: **Xep hang: Kha**

**BTCB.25: Cước điện thoại**

Viết chương trình C++ để tính cước điện thoại bàn cho một hộ gia đình với các thông số như sau:

Phí thuê bao bắt buộc là 25 nghìn.

1. 600 đồng cho mỗi gọi của 50 phút đầu tiên.
2. 400 đồng cho mỗi phút gọi của 150 phút tiếp theo.
3. 200 đồng cho bất kỳ mỗi phút nào sau 200 phút đầu tiên

**BTCB.26: Đọc tháng**

Viết chương trình nhập vào 1 số nguyên x ( 0 < x < 13 ) . In ra màn hình tên tiếng anh là tháng của số vừa nhập . Ví dụ:

Nhập : 7

Kết quả: Thang 7 la July

**BTCB.27: So Sánh**

Viết chương trình nhập vào 1 số nguyên n ( -1e9 < x < 1e9 ) . In ra màn hình kết quả so sánh của số đó với số 8 . Ví dụ:

Nhập : 7

Kết quả: So 7 nho hon 8 .

**BTCB.28: Tính điểm**

Viết chương trình C++ xếp hạng học lực của học sinh dựa trên các điểm bài kiểm tra, điểm thi giữa kỳ, điểm thi cuối kỳ. Nếu:

· Điểm trung bình >= 9.0 là hạng A.

· Điểm trung bình >=7.0 và < 9.0 là hạng B

· Điểm trung bình >=5.0 và < 7.0 là hạng C

· Điểm trung bình <5.0 là hạng F

**BTCB.29: Tính doanh số**

Cửa hàng của bạn nhận gửi bán sản phẩm cho một công ty khác và hưởng hoa hồng, với mức hoa hồng theo doanh số bán như sau:

* 5% nếu tổng doanh số nhỏ hơn hoặc bằng 100 triệu.
* 10% nếu tổng doanh số nhỏ hơn hoặc bằng 300 triệu.
* 20 % nếu tổng doanh số là lớn hơn 300 triệu.

Hãy viết chương trình để tính hoa hồng bạn sẽ nhận được dựa trên doanh số bán hàng với đầu vào là 1 số thực n - doanh số. ( 0 <= n <= 1e18 )

**BTCB.30: Kế Toán**

Giả sử bạn là kế toán cho một công ty. Bạn cần viết một chương trình C++ để nhập lương nhân viên, tính thuế thu nhập và lương ròng (số tiền lương thực sự mà nhân viên đó nhận được). Với các thông số sau (mình chỉ đưa ra các con số này cho dễ tính toán):

30% thuế thu nhập nếu lương là 15 triệu

20% thuế thu nhập nếu lương từ 7 đến 15 triệu

10% thuế thu nhập nếu lương dưới 7 triệu

**BTCB.31: In 10 ký tự**

Viết chương trình nhập vào 1 ký tự **c** nằm trong bảng chữ cái. In ra 10 ký tự c.

Ví dụ:

Nhập : **a**

=> kết quả: Ket qua la: **aaaaaaaaaa**

**BTCB.32: In từ 1 đến n**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số nguyên từ 1 đến n.

Ví dụ:

Nhập : **6**

=> kết quả: Ket qua la: **1 2 3 4 5 6**

**BTCB.33: In từ n về 1**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số nguyên từ n đến 1.

Ví dụ:

Nhập : **6**

=> kết quả: Ket qua la: **6 5 4 3 2 1**

**BTCB.34: In các số lẻ**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là lẻ tính từ 1 đến n ;

Ví dụ:

Nhập : **10**

=> kết quả: Ket qua la: **1 3 5 7 9**

**BTCB.35: In ngược các số lẻ**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là lẻ tính từ 1 đến n ;

Ví dụ:

Nhập : **10**

=> kết quả: Ket qua la: **9 7 5 3 1**

**BTCB.36: In các số chẵn**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là chẵn tính từ 1 đến n ;

Ví dụ:

Nhập : **10**

=> kết quả: Ket qua la: **2 4 6 8 10**

**BTCB.37: In ngược các số chẵn**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là chẵn tính từ n về 1 ;

Ví dụ:

Nhập : **10**

=> kết quả: Ket qua la: **10 8 6 4 2**

**BTCB.38: In dãy số lệch 3**

Cho dãy số: 1, 4, 7, 10, 13…….

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là thuộc dãy số trên mà nhỏ hơn bằng n ;

Ví dụ:

Nhập : **18**

=> kết quả: Ket qua la: **1 4 7 10 13 16**

**BTCB.39: In ngược dãy số lệch 3**

Cho dãy số: n, n-3, n-6, n- 9…. 1

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số là thuộc dãy số trên từ 1 đến n ;

Ví dụ:

Nhập : **18**

=> kết quả: Ket qua la: **18 15 12 9 6 3**

**BTCB.40: Tìm ước**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra các số từ 1 đến n là ước của n ;

Ví dụ:

Nhập : **6**

=> kết quả: **Cac uoc cua 6 la: 1 2 3 6 ;**

**BTCB.41: Đếm số lượng ước**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra 1 số nguyên - số lượng ước dương của n

Ví dụ:

Nhập : **6**

=> kết quả: **So luong uoc cua 6 la : 4 ;**

**BTCB.42: Tính tổng các ước**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra 1 số nguyên - tổng các ước dương của n

Ví dụ:

Nhập : **6**

=> kết quả: **Tong cac uoc cua 6 la : 12 ;**

**BTCB.43: Tính tổng dãy số**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 100 );**

In ra 1 số nguyên - tổng các số từ 1 đến n ;

Ví dụ:

Nhập : **5**

=> kết quả: **Tong tu 1 den 6 la: 15;**

**BTCB.43: Kiểm tra nguyên tố**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 1e6);**

In ra thông báo số n có phải là số nguyên tố không .

Ví dụ:

Nhập : **6**

=> kết quả: **So 6 khong phai la so nguyen to**

**BTCB.44: Kiểm tra số chính phương**

Viết chương trình nhập vào 1 số nguyên **n. ( 1 <= n <= 1e6);**

In ra thông báo số n có phải là số chính phương không .

Ví dụ:

Nhập : **Nhap n= 6**

=> kết quả: **So 6 khong phai la so chinh phuong**

**BTCB.45: Tìm độ dài ký**

Viết chương trình nhập vào 1 số nguyên **n** và 1 chuỗi s - có độ dài là n **.**

**( 1 <= n <= 1e6);**

In ra 1 số nguyên k - độ dài lớn nhất của các ký tự giống nhau liên tiếp trong s ;

Ví dụ:

Nhập :  **6 aabbbb**

=> kết quả: **4**

**\*Giải thích:** độ dài của của các ký tự liên tiếp giống nhau là 2 ( aa) và 4 (bbbb);

4 > 2 => đáp án: 4

**BTCB.46: Viết hoa**

Viết hoa là viết một từ có chữ cái đầu tiên là chữ cái viết hoa. Nhiệm vụ của bạn là viết hoa từ đã cho.

Lưu ý rằng trong quá trình viết hoa, tất cả các chữ cái ngoại trừ chữ cái đầu tiên vẫn không thay đổi.

**Đầu vào:** Một dòng duy nhất chứa một từ **s** không trống. Từ này bao gồm các chữ cái tiếng Anh viết thường và viết hoa. Độ dài của từ sẽ không vượt quá 1e3.

**Đầu ra:** Từ đã cho sau khi viết hoa.

Ví dụ: **konjac => Konjac**

**BTCB.47: Word**

Hà rất khó chịu khi nhiều người trên mạng trộn chữ hoa và chữ thường trong một từ. Đó là lý do tại sao anh ấy quyết định phát minh ra một tiện ích mở rộng cho trình duyệt yêu thích của mình, có thể thay đổi đăng ký các chữ cái trong mỗi từ để nó chỉ bao gồm các chữ cái viết thường hoặc ngược lại, chỉ bao gồm các chữ cái viết hoa. Tại đó, càng ít chữ cái càng tốt nên được thay đổi trong từ. Ví dụ: từ **HoUse** phải được thay thế bằng **house** và từ **ViP** thay bằng **VIP** . Nếu một từ chứa số lượng chữ cái viết hoa và viết thường bằng nhau, bạn nên thay thế tất cả các chữ cái bằng những chữ cái viết thường. Ví dụ: **maTRIx** nên được thay thế bằng **matrix** . Nhiệm vụ của bạn là sử dụng phương pháp đã cho trên một từ nhất định.

**Đầu vào**

Dòng đầu tiên chứa một từ **S** - nó bao gồm các chữ cái Latinh viết hoa và viết thường và có độ dài từ 1 đến 100 .

**Đầu ra**

In từ đã sửa **S**

**BTCB.48: Khôi Phục**

Viết chương trình nhập vào lần lượt 2 chuỗi **s** và **r**  có chiều dài bằng nhau.

Nhiệm vụ của bạn là kiểm tra xem chuỗi **r** có phải là chuỗi đảo ngược của **s** .

Nếu có in ra “YES”, ngược lại in “NO” ;

**Note**: Một chuỗi được gọi là đảo ngược nếu viết ngược chuỗi đó ta được 1 chuỗi giống hết chuỗi gốc. Ví dụ:

* Chuỗi “ba” là chuỗi đảo ngược của chuỗi “ab”
* chuỗi “bca” không là chuỗi đảo ngược của chuỗi “abc”

Ví dụ: **ba ab**

Nhập :  **YES**

**BTCB.49: Chuỗi đối xứng**

Viết chương trình nhập vào 1 chuỗi **s**. Kiểm tra xem nó có phải là chuỗi đối xứng hay không .

Nếu có in ra “YES”, ngược lại in “NO” ;

**Note**: Một chuỗi được gọi là đối xứng nếu đọc ngược , đọc xuôi thì vẫn là chuỗi đó. Ví dụ:

* Chuỗi “aba” ,”a” , “aaaa” là chuỗi đối xứng
* chuỗi “bca”,” hi” không là chuỗi đối xứng

Ví dụ: Nhập : **aba**

Kết quả :  **YES**

**BTCB.50: H và D**

Viết chương trình nhập vào 1 số nguyên **n** và 1 chuỗi s - có độ dài là n chỉ bảo gồm 2 ký tự ‘H’ và ‘D’ **.**

**( 1 <= n <= 100);**

Nhiệm vụ của bạn là so sánh số lượng ký tự ‘H’ và ‘Đ’ .

Nếu số ký tự ‘H’ lớn hơn hoặc bằng ‘D’, in “Ha”, ngược lại, in “Dat” ;

Ví dụ:

Nhập :  **6 HHHDDD**

=> kết quả: **Ha**

**BTCB.51: Tổng chữ số**

Viết chương trình nhập vào 1 số nguyên **a ( 1 <= a <= 1e9);**

Tỉnh tổng các chữ số của số đã nhập vào.

Ví dụ:

Nhập :  **123**

=> kết quả: **6**

***Giải thích****: vì 1 + 2 + 3 = 6*

**BTCB.52: Lucky Number**

Ta định nghĩa số may mắn là số chỉ gồm 6 hoặc 8 hoặc cả 6 và 8.

Ví dụ, 6, 8, 68, 86, 88,.. nhưng 67 , 3,5 .. thì không

Viết chương trình nhập vào 1 số nguyên **a ( 1 <= a <= 1e9);**

Kiểm tra xem nó có là số may mắn hay không?

Nếu đúng thì in YES, ngược lại, NO ;

Ví dụ:

Nhập :  **66**

=> kết quả: **YES**

**BTCB.53: In số**

Viết chương trình nhập vào 1 dãy số nguyên **a ( 1 <= a <= 1e9)** và in ra số đó, đến khi gặp số 0 thì dừng chương trình .

Ví dụ:

Nhập :  **1 2 3 4 1 0**

=> kết quả: **1 2 3 4 1 0**

**BTCB.54: Âm và Dương**

Viết chương trình nhập vào 1 dãy số nguyên **a ( 1 <= a <= 1e9)** đến khi gặp số 0 . Tính tổng các số dương và trung bình cộng các số âm trong dãy số đó theo định dạng dưới đây .

Ví dụ:

Nhập :  **1 -1 2 3 -1 0**

=> kết quả: **Tong cac so duong la: 6**

**TBC cac so am la: -1**

**BTCB.55 : In và check**

Viết chương trình nhập vào 1 dãy số nguyên **a ( 1 <= a <= 100)** đế.

Dùng vòng lặp do while in ra các số lẻ từ 1 đến a

Ví dụ:

Nhập :  **10**

=> kết quả: **1 3 5 7 9**

**// Từ đây là bài tập về mảng:**

**BTCB.56 : Tính Tổng Lẻ**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng! ( có thể âm)

In ra các phần tử ở vị trí lẻ trong mảng và tổng của chúng

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

=> kết quả: **Cac phan tu o vi tri le la: 2 3 5**

**Tong cua chung la : 10**

**BTCB.57 : Tính Tổng Chẵn**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng! ( có thể âm)

In ra các phần tử ở vị trí chẵn trong mảng và tổng của chúng

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

=> kết quả: **Cac phan tu o vi tri chan la: 1 3 4**

**Tong cua chung la : 8**

**BTCB.58 : In chẵn lẻ**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng! ( có thể âm)

In ra các phần tử là chẵn và các phần tử là lẻ trong mảng.

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

=> kết quả: **Cac phan tu chan la: 2 4**

**Cac phan tu le la: 1 3 3 5**

**BTCB.59 : Tính Tổng các số âm**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng! ( có thể âm)

In ra tổng các số âm có trong mảng

Ví dụ:

Nhập :  **6**

**1 2 3 -3 -2 -1**

=> kết quả: **Tong cac so am co trong mang la: -6**

**BTCB.60 : Tìm kiếm trong mảng**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Hai dòng tiếp theo: Dòng thứ nhất: nhập vào **n** phần tử của mảng! -

Dòng thứ hai: Nhập 1 số nguyên **k ( 1 <= k <= 1e5)**.

In ra **Yes** nếu số **k** có trong mảng, ngược lại in ra **No**

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

**6**

=> kết quả: **No**

**BTCB.61 : Đếm số lượng**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Hai dòng tiếp theo: Dòng thứ nhất: nhập vào **n** phần tử của mảng!

Dòng thứ hai: Nhập 1 số nguyên **k ( 1 <= k <= 1e6)**.

In ra số lần xuất hiện của **k** trong mảng .

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

**3**

=> kết quả: **So lan xuat hien cua 3 trong mang la: 2**

**BTCB.62 : In Ngược**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

In ra các phần tử của mảng từ cuối mảng về đầu mảng

Ví dụ:

Nhập :  **6**

**1 2 3 3 4 5**

=> kết quả: **5 4 3 3 2 1**

**BTCB.63 :Tổng các số dương và xóa số âm**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

Xóa các số âm có trong mảng

In ra mảng sau khi xóa và in ra tổng các số dương có trong mảng.   
 ( **note** : không cần sắp xếp lại mảng)

Ví dụ:

Nhập :  **6**

**1 2 3 -3 -2 1**

=> kết quả: **Tong so duong co trong mang la : 6**

**Mang sau khi xoa la: 1 2 3 1**

**BTCB.64 : Số lượng và TBC chia hết của 3**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

In ra số lượng phần tử chia hết cho 3, in ra các số đó và trung bình cộng của các số ấy!

Ví dụ:

Nhập :  **6**

**1 2 3 3 2 6**

=> kết quả: **Co 3 so chia het cho 3 la: 3 3 6**

**TBC cua cac so do la: 4**

**BTCB.65 : Sắp xếp tăng**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

In ra các phần tử của mảng theo thứ tự tăng dần .

Ví dụ:

Nhập :  **6**

**-1 2 3 1 2 3**

=> kết quả: **Mang sau khi sap xep tang la : -1 1 2 2 3 3**

**BTCB.66 : Sắp xếp giảm**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e6) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

In ra các phần tử của mảng theo thứ tự giảm dần . .

Ví dụ:

Nhập :  **6**

**-1 1 2 3 1 2 3**

=> kết quả: **Mang sau khi sap xep tang la : 3 3 2 2 1 -1**

**BTCB.67 : Thay thế**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e6) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng!

Thay thế các số âm trong mảng bằng số 0 .

In ra các phần tử của mảng sau khi đã áp dụng thay đổi trên .

Ví dụ:

Nhập :  **6**

**1 -2 3 -1 2 -6**

=> kết quả: **Mang sau khi thay the la: 1 0 3 0 2 0**

**BTCB.68 : Chèn phần tử**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e6) - độ dài mảng**.

Dòng tiếp theo, nhập vào n phần tử của mảng!

Sắp xếp mảng theo thứ tự giảm dần.

Nhập 1 số nguyên **k** bất kì . Thực hiện chèn số nguyên **k**  đó vào mảng sao cho mảng vẫn được sắp xếp giảm dần.

In ra mảng trước và sau khi chèn số nguyên **k** đó.

Ví dụ:

Nhập :  **6**

**-1 2 3 1 2 3**

**Nhap so can chen la : 7**

=> kết quả: **Mảng trước khi chèn la : 3 3 2 2 1 -1**

**Mảng sau khi chèn la : 7 3 3 2 2 1 -1**

**BTCB.69 : Check hoan vi**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên dương - các phần tử của mảng.

kiểm tra xem các phần tử của mảng có phải là 1 hoán vị của n hay không .

Nếu có, in **Yes ,** ngược lại in **No.**

***note:*** *Dãy gồm n số nguyên được gọi là một hoán vị nếu nó chứa tất cả các số nguyên từ 1 đến n đúng một lần.*

***ex:***  *1 2 3 hay 3 1 2… là 1 hoán vị của 3 nhưng 1 2 4 hay 1 3 4…. thì không*

Ví dụ:

Nhập :  **6**

**3 4 5 1 2 6**

=> kết quả: **YES**

**BTCB.70: Array Palindrome**

Như ta đã tìm hiểu thì chuỗi **Palindrome**( hay còn gọi là chuỗi đối xứng ) là các chuỗi mà khi ta đọc ngược hay đọc xuôi đều cho 1 kết quả, chẳng hạn như các chuỗi: **aaaa, aba, level**... là các chuỗi Palindrome.

Từ định nghĩa trên, trong vấn đề này, ta sẽ kiểm tra 1 mảng xem mảng đó có phải là mảng Palindrome hay không .

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1e5) - độ dài mảng**.

Dòng tiếp theo, nhập vào n số nguyên - các phần tử của mảng!

Nếu mảng vừa nhập là mảng Palindrome, in ra **Yes**; ngược lại in **No**.

Ví dụ:

Nhập :  **6**

**1 2 3 3 2 1**

=> kết quả: **Yes**

**\*BTCB.71: Tính số Fibonacci**

Viết chương trình nhập vào 1 số nguyên **n ( 1 <= a <= 1000)**

Hãy dùng mảng tính số Fibonacci thứ n.

In ra kết quả tìm được

Ví dụ:

Nhập :  **38**

Kết quả: **So Fibonacci thu 38 la:** **39088169**

**\*BTCB.72: Mảng đánh dấu:**

Viết chương trình nhập vào 2 số nguyên **n,q ( 1 <= n; q <= 1e5)**

Dòng thứ 2, nhập n số nguyên dương - các phần tử của mảng **a**

q dòng tiếp theo, mỗi dòng nhập vào 1 số nguyên dương .

Với mỗi dòng đó, kiểm tra xem nó đã xuất hiện trong mảng **a**  hay chưa .

Nếu đã xuất hiện, in **Yes**, ngược lại in **No**

Ví dụ:

Nhập :  **6 3**

**1 2 4 4 8 6**

**1**

**2**

**3**

Kết quả: **Yes**

**Yes**

**No**

***Giải thích:*** Mảng nhập vào là:  **1 2 4 4 8 6**

Với truy vấn là **1 :** vì số **1** đã xuất hiện trong mảng nên kết quả là **Yes**

Với truy vấn là **2 :** vì số **2** đã xuất hiện trong mảng nên kết quả là **Yes**

Với truy vấn là **3 :** vì số **3** không xuất hiện trong mảng nên kết quả là **No**

***Note***

**Từ giờ trở đi, các bài tập dưới đây ngoài việc làm đúng thì mọi người viết bằng chương trình con mới là đạt yêu cầu**

**BTCB.73: Thay thế trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình thay thế các phần tử âm trong **arr**  thành số 0 và in **arr** sau khi đã sửa. ( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**-5 -7 3**

**-1 -2 4**

Kết quả: **Ma tran sau khi sua la:**

**0 0 3**

**0 0 4**

**BTCB.74: Tính tổng các phần của chia hết cho 5**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình tính in ra các số chia hết cho 5 và tổng các phần tử đó trong **arr** ( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ: nhập :

**3 4**

**10 4 8 2**

**5 6 4 9**

**20 3 8 1**

Kết quả: **Cac so chia het cho 5 trong ma tran la: 10 5 20**

**Tong cua cac so do la: 35**

**BTCB.75: Tính tổng các số dương trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình tính tổng các phần tử dương trong **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 3**

**1 2 4**

Kết quả: **22**

**\*BTCB.76: Tính tổng các số giống nhau trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình tính tổng các phần tử giống nhau trong **arr**

( **0** < **n,m** <= **1e4**; -**1e2**<= **arr[i] <= 1e3**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**1 2 2**

Kết quả: **18**

*Giải thích: 7 + 7 + 2 + 2 = 18*

**BTCB.77: Tìm số lớn nhất trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình in ra số là số lớn nhất trong **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

Kết quả: **So lon nhat trong ma tran la: 8**

**BTCB.78: Tìm số bé nhất trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình in ra số là số bé nhất trong **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

Kết quả: **So be nhat trong ma tran la: 4**

**BTCB.79: Tìm số lớn thứ 2 trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình in ra số là số lớn thứ 2 trong **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

Kết quả: **So lon thu 2 trong ma tran la: 7**

**BTCB.80: Tìm số bé thứ 2 trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình in ra số là số lớn nhất trong **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

Kết quả: **So be thu 2 trong ma tran la: 5**

**BTCB.81: Tìm số nguyên tố trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím. Bạn hãy viết chương trình in ra các số là số nguyên tố và tổng của chúng trong **arr**  ( **0** < **n,m** <= **1e4**; 0 <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

Kết quả: **Cac so nguyen to trong ma tran la: 5 7 7**

**Tong cua chung la: 19**

**BTCB.82: Tính tổng của hàng bất kỳ trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím.

Sau đó, nhập thêm 1 số nguyên **k** bất kỳ ( **0 < k <= n** )

Bạn hãy viết chương trình in tổng của các phần tử trong hàng thứ **k** của **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ: Nhập:

**2 3**

**5 7 7**

**4 6 8**

**1**

Kết quả: **Tong cac phan tu cua hang thu 1 la: 19**

**BTCB.83: Tính tổng của cột bất kỳ trong ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím.

Sau đó, nhập thêm 1 số nguyên **k** bất kỳ ( **0 < k <= m** )

Bạn hãy viết chương trình in tổng của các phần tử trong cột thứ **k** của **arr**

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**2 3**

**5 7 7**

**4 6 8**

**1**

Kết quả: **Tong cac phan tu cua cot thu 1 la: 9**

**BTCB.84: In biên của ma trận**

Cho một mảng 2 chiều **arr** các số nguyên **n** hàng **m** cột được nhập từ bàn phím.

Bạn hãy viết chương trình in các phần tử là biêncủa **arr**  theo thứ tự xuất hiện từ trái qua phải, trên xuống dưới

( **0** < **n,m** <= **1e4**; -**1e6** <= **arr[i] <= 1e6**  )

Ví dụ:

Nhập:

**3 3**

**5 7 7**

**4 6 8**

**3 4 6**

Kết quả: **Cac phan tu la bien cua ma tran la: 5 7 7 4 8 3 4 6**

***Struct***

***Note: các bài từ 86 trở đi, chỉ cần mọi người viết chương trình con + đúng thứ tự yêu cầu bài toán + dữ liệu bắt thực tế +là đạt yêu cầu ( biểu diễn chương trình dễ nhìn - hiểu)***

**BTCB.85: Nhập hiện Struct**

Viết chương trình nhập vào 1 danh sách gồm **n** quyển sách của cửa hàng sách GenZ gồm: **Tên sách, tác giả** .**(0< n <= 100)**. Yêu cầu: In ra màn hình danh sách đã nhập .

**BTCB.86: Cộng 2 phân số**

Viết chương trình C++ để cộng hai phân số và hiển thị kết quả. Chương trình của bạn sẽ nhắc người dùng nhập phân số 1 và phân số 2. Tử số và mẫu số được nhập một cách riêng rẽ bởi khoảng trống, như định dạng dưới đây:

Input:

**Nhap phan so thu nhat:**

**Tu so la: 1**

**Mau so la: 3**

**Nhap phan so thu hai:**

**Tu so la: 2**

**Mau so la: 5**

Output:

**Hai phan so da nhap la:1 / 3 va 2 / 5**

**Tong 2 phan so do la : 11 / 15**

**BTCB.87: In danh sách thi lại**

Viết chương trình nhập vào **họ tên** và **điểm thi** của các sinh viên của 1 lớp đến khi **không nhập nữa thì dừng**. In ra danh sách các sinh viên phải thi lại nếu điểm thi dưới 5.0 .

**BTCB.88: Sắp xếp Struct**

Viết chương trình nhập vào 1 danh sách của **n** sinh viên lớp GenZ gồm: **họ tên, năm sinh, điểm thi** vào trường **(0< n <= 100)**. Yêu cầu:

* Đưa ra danh sách này ra màn hình.
* Sắp xếp danh sách theo chiều giảm dần của điểm thi và đưa ra màn hình kết quả sau khi đã sắp xếp .
* Sắp xếp danh sách theo chiều tăng dần của năm sinh và đưa ra màn hình kết quả sau khi đã sắp xếp .

**BTCB.89: Min max trong Struct**

Viết chương trình nhập vào 1 danh sách của **n** sinh viên lớp GenZ gồm: họ tên, năm sinh, điểm thi vào trường **(0< n <= 100)**. Yêu cầu:

* Đưa ra danh sách này ra màn hình.
* Đưa ra màn hình thông tin danh sách sinh viên có tuổi lớn nhất.
* Đưa ra màn hình thông tin danh sách sinh viên có tuổi bé nhất.

**BTCB.90: Remove phần tử**

Viết chương trình nhập vào 1 danh sách của **n** sinh viên lớp GenZ gồm: họ tên, năm sinh, giới tính **(0< n <= 100)**. Yêu cầu:

* Đưa ra danh sách này ra màn hình.
* Đưa ra màn hình thông tin danh sách sinh viên nữ sinh sau năm 2002.
* Xóa các sinh viên nữ dưới 19 tuổi rồi đưa ra danh sách sau khi xóa.

**BTCB.91: Weather**

Viết chương trình thực hiện các yêu cầu dưới đây:

1. Nhập thông tin từ bàn phím về tình hình thời tiết trong ngày của khu vực. Mỗi bản ghi là 1 bản ghi gồm các trường:

* **Ngày** ( thỏa mãn 0 => 31) ;

**Tháng** ( thỏa mãn 1 => 12) ;

**Năm**  ( thỏa mãn 1990 => 2021) ;

* **Địa điểm:** là xâu ký tự dài không quá 30 ;
* **Lượng mưa:**  là giá trị thực ;
* **Nhiệt độ:**  là giá trị nguyên ;

Số lượng bản ghi không biết trước nhưng sẽ nhỏ hơn 100, dấu hiệu kết aaaaaathúc là ngày bằng 0 ;

In danh sách đã nhập ra màn hình

1. Xác định nhiệt độ cao nhất , thấp nhất là bao nhiêu. Xảy ra ở đâu, ngày nào?
2. Đưa ra màn hình lượng mưa trung bình trong ngày của danh sách.

**BTCB.92: Commodity**

Cho 1 danh sách hàng hóa gồm: **tên mặt hàng**, **số lượng**, **đơn giá** ( giá của 1 đơn vị mặt hàng ), **thành tiền** ( bằng số lượng \* đơn giá ) .

Viết chương trình thực hiện các công việc sau:

1. Nhập dữ liệu từ bàn phím cho danh sách trên

< Kết thúc nhập khi tên mặt hàng bằng ’\*’ , số lượng không quá 100 ,

thành tiền phải tự tính, không được nhập >

1. Đưa ra màn hình danh sách đã nhập trên.
2. Đưa ra màn hình thông tin sản phẩm của hàng hóa có số lượng ít hơn 6 .
3. Nhập vào 1 chuỗi **s** . In ra thông tin hàng hóa có tên trùng với chuỗi đã nhập.
4. Đưa ra danh sách hàng hóa sau khi sắp xếp theo chiều tăng dần của thành tiền .

**BTCB.93: Bố trí tối ưu**

Có **n** cuộc họp đánh số từ **1** đến **n** đăng ký làm việc tại một phòng hội thảo. Cuộc họp **i** cần được bắt đầu ngay sau thời điểm **si** và kết thúc tại thời điểm **fi­**. Hỏi có thể bố trí phòng hội thảo phục vụ được nhiều nhất bao nhiêu cuộc họp, sao cho khoảng thời gian làm việc của hai cuộc họp bất kỳ là không giao nhau.

### Input

* Dòng đầu tiên chứa số nguyên dương n ( n <= 10000)
* Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên dương **si, fi**

**(si < fi <= 32000) ( 1 <= i <= n).**

### Output

* In ra 1 số nguyên K duy nhất: số các cuộc họp tối đa được chấp nhận phục vụ

**Ví dụ:**

Input: Output:

**5 3**

**7 9**

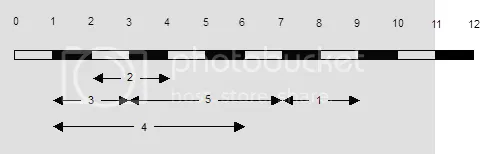
**2 4**

**1 3**

**1 6**

**3 7**

Giải thích: giải pháp tối ưu sẽ là 3 cho các mốc thời gian sau : 1-3 , 3-7, 7-9

****