

Nama Samaran (A1)



Roronoa Zoro sedang tersesat ke sebuah kota bawah laut bernama Bikini Bottom. Zoro membutuhkan bantuan untuk membuat nama samaran untuk dirinya. Bantu Zoro membuat sebuah program untuk membuat nama samaran yang menukar karakter pertama antara nama depan dan nama belakang.

Input

Roronoa Zoro

Output

Zoronoa Roro

Test Case

input	output
Spongebob Squarepants	Spongebob Squarepants
Naruto Uzumaki	Uaruto Nzumaki
Monkey Luffy	Lonkey Muffy
Nico Robin	Rico Nobin

Musuh Zoro (A1)



Roronoa Zoro merupakan anggota terkuat kedua setelah Luffy di kru topi jerami. Posisinya sebagai orang terkuat kedua di kru membuat dia selalu melawan orang terkuat kedua pula di pihak lawan. Jika diketahui *power-level* dari lawan-lawan kru topi jerami. Tentukan kekuatan musuh yang akan dilawan oleh Zoro.

Cara input list

```
user_input = [int(x) for x in input().split()]
```

Input 1

42 12 33 21 50

Output 1

42

Input 2

100 545 123 99 212

Output 2

212

Input 3

10 20 30 40 50

Output 3

40

Bantu Rias (A2)

Perusahaan Rias Gremory akhir akhir ini mengalami kesusahan dalam pencatatan portofolio pembelian dan penjualan saham mereka. Oleh karena itu, Issei, seorang karyawan diperintahkan Rias Gremory untuk membuat sebuah program yang menghitung keuntungan atau kerugian dari penjualan saham mereka.

Input: input line pertama terdiri dari berapa testcase saham yang akan dihitung. line selanjutnya adalah harga beli dan harga jua per 1 lot yang dipisahkan spasi per testcase

Output: Total keuntungan atau kerugian yang didapat Rias Gremory

Asumsikan mereka selalu bertransaksi 100 lot setiap testcase

Input

```
3
1200 1500
2000 3000
5500 5000
```

Output

```
80000
```

Input

```
3
1000 1900
2000 1300
5500 4000
```

Output

```
-170000
```

Input

```
3
5400 100
50 5050
300 200
```

Output

```
-400
```

Pengeluaran Bulanan (B1)

Arya sedang kebingungan di akhir bulan karena uang tabungannya yang habis. Untuk itu, maka di bulan berikutnya Arya mencatat setiap pengeluarannya untuk nantinya di rekap dan ia dapat mengetahui pengeluaran terbesarnya berapa.

Input:

n: [1...n] banyaknya pengeluaran yang akan masuk ke dalam list

L: pengeluaran (100 = makan, 200 = bensin,...)

Input

5
100 jajan
700 makan
850 bensin
400 lain - lain
200 listrik

Output

850 bensin

Input 2

3
20000 warpad
22000 sate padang
17000 oti

Output 2

22000 sate padang

Input 3

4
1000 parkir oti
2000 parkir pasundan
5000 soto bu bambang
500 sumbangan

Output 3

5000 soto bu bambang

To Do List (B2)

Tobi kesusahan mengingat kegiatannya sehari-hari. Bantulah Tobi agar bisa mengingat kegiatannya setiap hari dengan membuat to-do-list.

Input:

n : integer [1..50] (banyaknya kegiatan yang dicatat)

L : list of string (list kegiatan, banyaknya kegiatan adalah n)

Output:

isi dari list kegiatan yang dimasukkan.

note: gunakan teknik rekursif!

Input 1

```
3
mancing
makan
main
```

Output 1

```
['mancing', 'makan', 'main']
```

Pembahasan contoh 1:

3 adalah nilai n, yaitu banyaknya elemen yang akan dimasukkan ke list.

makan, mancing, main adalah elemen list.

['mancing', 'makan', 'main'] adalah hasil print dari list yang dibuat.

Input 2:

```
51
```

Output 2:

```
Tobi tidak bisa bekerja terlalu keras!
```

Pembahasan contoh 2:

masukan n di luar batas.

Input 3:

```
0
```

Output 3:

```
Tobi tidak ingin bermalas-malasan!
```

Pembahasan contoh 3:
masukan n di luar batas.

Test Case

test case 1:

Input:

51

Output:

Tobi tidak bisa bekerja terlalu keras

test case 2:

Input:

-99

Output:

Tobi tidak ingin bermalas-malasan!

test case 2:

Input:

-99

Output:

Tobi tidak ingin bermalas-malasan!

test case 3:

Input:

5
tidur
latihan
kerja
main
kerja

Output:

['tidur', 'latihan', 'kerja', 'main', 'kerja']

Seleksi Akatsuki(B2)

Pain memiliki beberapa kandidat untuk dijadikan member baru akatsuki. Kandidat-kandidat tersebut memiliki nilai hasil misi pertama yang disimpan di dalam list. Jika nilai seorang kandidat lebih dari nilai x , kandidat tersebut lolos. Bantulah Pain untuk menghitung berapa banyak kandidat yang lolos.

Input:

x : integer ≤ 100 (nilai minimum untuk lolos)

L : list of integer (list yang berisi nilai-nilai para kandidat)

output:

banyaknya kandidat yang lolos

Keterangan:

Untuk menerima input list, gunakan:

```
inp = list(map(int, input().split(' ')))
```

```
# Bila diberi input 10 9 8 7 6 5, inp = [10, 9, 8, 7, 6, 5]
```

Contoh Input 1:

```
60
70 100 50 35 90
```

Contoh Output 1:

```
3
```

Pembahasan contoh 1:

Nilai x adalah 60. Dari 5 kandidat, ada 3 yang lebih dari atau sama dengan x , yaitu 70, 100, dan 90.

Contoh Input 2:

```
85
80 79 81
```

Contoh Output 2:

```
0
```

Pembahasan contoh 2:

Nilai x adalah 80. Dari semua kandidat, tidak ada yang mencapai nilai 85, maka output 0.

Test Case

test case 1:

Input:

```
50
30 50
```

Output:

1

test case 2:

Input:

70
30 50 20 60 40 10

Output:

0

test case 3:

Input:

70
0

Output:

0

test case 4:

Input:

60
90 25 69 59 70

Output:

3

test case 5:

Input:

90
90 99 100 100 91

Output:

5

Tolong Ucup (C1)

Ucup merupakan mahasiswa yang mempunyai bisnis berupa berjualan permen dengan sistem pembayaran KEJUJURAN. Setiap hari dia menyimpan dagangannya di samping tangga gedung B kampus tercinta. Ucup menjual permennya seharga 5000 rupiah per satu permen. Namun, jika membeli 5 sekaligus hanya perlu membayar 20000 rupiah. Ia kebingungan setiap melakukan penghitungan. Buatlah algoritma untuk membantu ucup menghitung berapa banyak penjualan satuan dan 5 buah sekaligus! (Diasumsikan semua pembeli jujur)

Input

A (Banyak uang yang didapat)

Output

X (Banyak pembeli satuan)

Input 1

50000

Output 1

12

Input 2

60000

Output 2

15

Input 2

15000

Output 2

3

CodeSplit (C1)

Senkuu adalah seorang programmer, dia diminta tolong untuk memecahkan kode mulai dari sekumpulan angka 0-9 bahkan huruf. Bantu dia untuk memotong kode menjadi beberapa bagian sesuai Key untuk bisa di terjemahkan lebih mudah.

Input :

C : string[1..80]

X : integer [1..8]

Output :

List dari kode yang sudah di potong-potong

Note : Panjang Code (C) Harus bisa habis dibagi Key (X)

Gunakan append agar dapat input string ke dalam list

Input

1111000010100101
4

Output

['1111', '0000', '1010', '0101']

Input 2

1111000010100101
5

Output 2

Bukan code ini mah

Penjelasannya, panjang C input 2 adalah 16 sedangkan Key nya adalah 5, 16 tidak habis dibagi 5, jadi ada angka tersisa jika dipotong.

Input 3

HALO1234TESTAJAAYAAA
4

Output 3

['HALO', '1234', 'TEST', 'AJAA', 'YAAA']

Test Case

Input 1

0111001101110101011100110110000101101000011010100110100101110010
8

Output 1

['01110011', '01110101', '01110011', '01100001', '01101000', '01101010', '01101001',
'01110010']

Input 2

0110110101100001011011000110010101110011011000110110111101101011
2

Output 2

['01', '10', '11', '01', '01', '10', '00', '01', '01', '10', '11', '00', '01', '10', '01', '01', '01', '11', '00',
'11', '01', '10', '00', '11', '01', '10', '11', '11', '01', '10', '10', '11']

Input 3

11511197108107111115117115971049810311697110106103
3

Output 3

Bukan code ini mah

Input 4

U3VzYWggYmFuZ2V0IHdvaSBiaWtpbiBzb2FsLCB5YW9sb2guLiBnaW1hbmEgY29iYSwgc3VydSBtaWtpciwgdHAgaZ3BwLCBsb3ZlIHUgOjM=
1

Output 4

['U', '3', 'V', 'z', 'Y', 'W', 'g', 'g', 'Y', 'm', 'F', 'u', 'Z', '2', 'V', '0', 'I', 'H', 'd', 'v', 'a', 'S', 'B', 'i', 'a',
'W', 't', 'p', 'b', 'i', 'B', 'z', 'b', '2', 'F', 's', 'L', 'C', 'B', '5', 'Y', 'W', '9', 's', 'b', '2', 'g', 'u', 'L', 'i', 'B',
'n', 'a', 'W', '1', 'h', 'b', 'm', 'E', 'g', 'Y', '2', '9', 'i', 'Y', 'S', 'w', 'g', 'c', '3', 'V', 'y', 'd', 'S', 'B', 't', 'a',
'W', 't', 'p', 'c', 'i', 'w', 'g', 'd', 'H', 'A', 'g', 'Z', '3', 'B', 'w', 'L', 'C', 'B', 's', 'b', '3', 'Z', 'l', 'I', 'H', 'U',
'g', 'O', 'j', 'M', '=']

Input 5
DAHCAPEAKU 2

Output 5
['DA', 'HC', 'AP', 'EA', 'KU']

Odd First (C2)

Pak Dengklek memiliki beberapa pekerjaan dengan prioritas pengerjaan (p) berupa bilangan integer yang berbeda-beda dengan urutan dari prioritas terbesar ke terkecil (descending)

Secara mendadak, bos Pak Dengklek mempercepat deadline dari semua pekerjaan dengan prioritas ganjil, sehingga Pak Dengklek harus menyelesaikan pekerjaan dengan prioritas ganjil terlebih dahulu. Karena Pak Dengklek sangat sibuk, Pak Dengklek meminta bantuan anda untuk mengurutkan pekerjaannya agar pekerjaan dengan prioritas ganjil dan terbesar dikerjakan terlebih dahulu

Batasan Input

Satu baris berisikan deretan bilangan p ($1 \leq p \leq 100$) yang merupakan prioritas masing-masing pekerjaan Pak Dengklek

Batasan Output

Satu baris berisikan deretan bilangan integer dengan jumlah sama dengan input yang merupakan prioritas pekerjaan yang telah diurutkan dengan urutan:

1. Prioritas pekerjaan ganjil, lalu genap
2. Prioritas pekerjaan terbesar hingga terkecil

Input

contoh input
10 9 8 7 6 5

Output

contoh output
[9, 7, 5, 10, 8, 6]

Penjelasan:

Dari prioritas [10, 9, 8, 7, 6, 5], prioritas yang ganjil urut dari yang tertinggi hingga terendah adalah [9, 7, 5], sedangkan prioritas yang genap urut dari yang tertinggi hingga terendah adalah [10, 8, 6]

Karena pekerjaan dengan prioritas ganjil harus dikerjakan terlebih dahulu, pekerjaan [9, 7, 5] di outputkan terlebih dahulu, lalu dilanjutkan dengan pekerjaan [10, 8, 6]

Keterangan:

Untuk menerima input list, gunakan:

```
inp = list(map(int, input().split(' ')))  
# Bila diberi input 10 9 8 7 6 5, inp = [10, 9, 8, 7, 6, 5]
```

Untuk output list, langsung print list yang ingin ditampilkan

```
print([1, 2, 3, 4, 5, 6, 7])
```

Test case

Input 1

10 9 8 7 6 5

Output 1

[9, 7, 5, 10, 8, 6]

Input 2

54 43 32 21 10 8 6 4 2 1

Output 2

[43, 21, 1, 54, 32, 10, 8, 6, 4, 2]

Input 3

10 8 4 2

Output 3

[10, 8, 4, 2]

Input 4

100 94 93 85 83 82 81 77 74 65 43 22 22 21 10 7 4 3 2 1

Output 4

[93, 85, 83, 81, 77, 65, 43, 21, 7, 3, 1, 100, 94, 82, 74, 22, 22, 10, 4, 2]

Input 5

15 13 11 9 7 5 3 1

Output 5

[15, 13, 11, 9, 7, 5, 3, 1]

Permen Manis (C2)

Author : Benhard Sim

Deskripsi :

budi dan anto masing-masing mempunya n buah permen dengan tingkat kemanisan yang berbeda. budi menyukai permen yang manis tetapi anto tidak begitu menyukai permen yang manis, mereka berdua pun sepakat untuk saling bertukar permen sehingga tingkat kemanisan permen yang mereka miliki sesuai dengan kesukaan mereka masing-masing. tetapi mereka hanya dapat menukar permen pada urutan yang sama untuk setiap satu pertukaran maka dikatakan mereka telah melakukan satu kali **transaksi**. Berapakah jumlah transaksi minimal yang di perlukan sehingga **semua** permen yang dimiliki anto dan dimiliki budi sesuai dengan preferensi mereka masing-masing

format input

baris pertama mengandung adalah integer n yaitu panjang dari array.
terdapat 2 baris selanjutnya dengan n buah integer, baris pertama merupakan kumpulan permen budi dengan tingkat kemanisannya, baris kedua merupakan kumpulan permen anto dengan kemanisannya.

note :

gunakan algoritma ini untuk melakukan input

input untuk panjang array

```
n = int(input())
```

input array budi dan anto

```
budi = list(map(int,input().strip().split()))[:n]
```

```
anto = list(map(int,input().strip().split()))[:n]
```

format output

sebuah integer yang memberikan nilai transaksi minimum yang diperlukan

test case

input

input 1
6 5 6 3 15 20 11 10 11 1 3 7 1

output

Output 1
2

input 1
3 5 7 1 10 3 2

output

Output 1
2

test case hidden

input 1
6 1 1 1 1 1 1 1 1 1 1 1 1

output

Output 1
0

input 1
6 1 2 3 4 5 6 6 5 4 3 2 1

output

Output 1
3

input 1
5 3 2 1 3 2 1 1 2 3 1 2 3

output

Output 1
