

Time limit	1 s
Memory limit	64 MB

**Buat program dengan nama antre.c!**

Sebuah taman bermain bernama **Labprosotan Theme Park** memiliki *customer service center* untuk melayani keluhan pengunjung. Namun, besarnya gaji petugas bagian pengantre membuat pengelola sebal. Mereka berpikir, "Kenapa bayar orang kalau kita bisa menggunakan mesin antre canggih?"

Sayangnya, karma buruk pun datang! Mesin itu mengalami *bug* aneh yang menyebabkan terjadinya duplikat nomor antre. Kini pengunjung harus berurusan dengan duplikasi nomor antrean.

Tuan Ray dan Anda merupakan konsultan IT yang ditugaskan membantu Labprosotan Theme Park. Bantu Tuan Ray memperbaiki mesin antre absurd ini sebelum pengelola Labprosotan diamuk massa.

Format Masukan

Sebuah **array of integer** dengan ukuran **n** terurut secara menaik (*ascending*).

Format Keluaran

*Array of integer* yang dipisahkan oleh spasi dan diakhiri *newline* pada ujung baris.

Batasan

0<n<100

Contoh Input 1

Masukan:

10

1 2 3 4 5 5 6 7 8 8

Keluaran:

1 2 3 4 5 6 7 8

Penjelasan

Terdapat duplikat pada nomor antrean **5** dan **8** sehingga harus dihapus duplikatnya.

Contoh Input 2

Masukan:

6

20 20 20 21 40 40

Keluaran:

20 21 40

Penjelasan

Terdapat duplikat pada nomor antrean **20** dan **40** sehingga harus dihapus duplikatnya.

Contoh Input 3

Masukan:

8

90 100 100 100 100 100 120 120

Keluaran:

90 100 120

Penjelasan

Terdapat duplikat pada nomor antrean **100** dan **120** sehingga harus dihapus duplikatnya.


# Petunjuk / Hint

Tidak perlu membuat *array* (larik) dinamis. Soal ini dapat diselesaikan dengan larik statis. **Tidak ada spasi pada ujung keluaran.**

## Catatan

Penulisan harus diakhiri dengan *newline* (*\n*) pada ujung jawaban. Pastikan format masukan dan keluaran mengikuti format yang diminta secara **persis**.



 [antre.c](https://antre.c)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.59 MB
2	10	Accepted	0.00 sec, 1.63 MB
3	10	Accepted	0.00 sec, 1.51 MB
4	10	Accepted	0.00 sec, 1.54 MB
5	10	Accepted	0.00 sec, 1.60 MB
6	10	Accepted	0.00 sec, 1.66 MB
7	10	Accepted	0.00 sec, 1.71 MB
8	10	Accepted	0.00 sec, 1.71 MB
9	10	Accepted	0.00 sec, 1.66 MB
10	10	Accepted	0.00 sec, 1.70 MB

Time limit	1 s
Memory limit	64 MB

Pada hari Minggu kemarin, Pak Fibonacci baru saja menemukan suatu deret bilangan yang menarik, di mana setiap bilangan merupakan jumlah dari dua bilangan sebelumnya. Ia memberikan nama deret Fibonacci untuk penemuan ini.

Pak Fibonacci berniat menantangmu untuk mencari jumlah dari bilangan-bilangan ganjil/genap dalam deret Fibonacci hingga suku ke-**n** (0-indexed). Pak Fibonacci juga ingin melihat array berisi susunan bilangan pada deret Fibonacci hingga suku ke-**n**.

Buatlah **ganjilgenap.c** dengan memanfaatkan TabInt dari [array.c](#) dan [array.h](#) yang telah kamu buat sebelumnya!

### Format Masukan

Dua buah angka integer **odd** dan **n**

Batasan:

0 <= **odd** <= 1  
0 <= **n** <= 46

### Format Keluaran

- Sebuah angka integer berupa jumlah dari bilangan-bilangan ganjil (apabila **odd** = 1) atau bilangan-bilangan genap (apabila **odd** = 0) dalam deret Fibonacci hingga suku ke-**n**
- Sebuah list susunan bilangan pada deret Fibonacci hingga suku ke-**n**

### Contoh Input dan Output

Input	Output	Penjelasan
1 10		Deret Fibonacci yang terbentuk adalah 0, 1, 1, 2, 3, 5
5	[0,1,1,2,3,5]	Bilangan ganjil: 1, 1, 3, 5 Jumlah bilangan ganjil: 1 + 1 + 3 + 5 = 10
0 10		Deret Fibonacci yang terbentuk adalah 0, 1, 1, 2, 3, 5, 8, 13
7	[0,1,1,2,3,5,8,13]	Bilangan genap: 0, 2, 8 Jumlah bilangan genap: 0 + 2 + 8 = 10
1 99		Deret Fibonacci yang terbentuk adalah 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55
10	[0,1,1,2,3,5,8,13,21,34,55]	Bilangan ganjil: 1, 1, 3, 5, 13, 21, 55 Jumlah bilangan ganjil: 1 + 1 + 3 + 5 + 13 + 21 + 55 = 99



[ganjilgenap.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.61 MB
2	10	Accepted	0.00 sec, 1.72 MB

No	Score	Verdict	Description
3	10	Accepted	0.00 sec, 1.72 MB
4	10	Accepted	0.00 sec, 1.61 MB
5	10	Accepted	0.00 sec, 1.65 MB
6	10	Accepted	0.00 sec, 1.72 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.68 MB
9	10	Accepted	0.00 sec, 1.65 MB
10	10	Accepted	0.00 sec, 1.63 MB

Time limit	1 s
Memory limit	64 MB

Buatlah ADT listMatkul.c yang merupakan implementasi dari [listMatkul.h](#)

Berikut adalah file [boolean.h](#)

C

 [listMatkul.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.65 MB
2	10	Accepted	0.00 sec, 1.58 MB
3	10	Accepted	0.00 sec, 1.61 MB
4	10	Accepted	0.00 sec, 1.72 MB
5	10	Accepted	0.00 sec, 1.72 MB
6	10	Accepted	0.00 sec, 1.68 MB
7	10	Accepted	0.00 sec, 1.60 MB
8	10	Accepted	0.00 sec, 1.63 MB
9	10	Accepted	0.00 sec, 1.60 MB
10	10	Accepted	0.00 sec, 1.70 MB