

🎀 Hello Kitty dan Pita Keberuntungan 🎀

Oleh: Dimas Setiaji

| | |
|--------------|--------|
| Time Limit | 1 s |
| Memory Limit | 256 MB |

Kitty White adalah seorang kolektor pita warna-warni dari Desa Sanrio Puroland. Setiap pita yang ia miliki memiliki kode angka unik yang mewakili warna dan makna keberuntungannya.

Suatu sore, Kitty duduk di ruang koleksinya dan mulai memikirkan semua kemungkinan kombinasi pita yang bisa ia susun. Ia ingin mencoba berbagai susunan pita, tetapi hanya tertarik pada kombinasi yang jumlah elemennya genap. Menurut kepercayaan lokal, kombinasi genap membawa keberuntungan dan harmoni dalam hidupnya.

Kitty mendefinisikan kombinasi pita sebagai subsekuens dari pita-pita yang ia miliki. Artinya, ia bisa memilih sebagian pita dalam urutan semula, tanpa harus berurutan penuh. Bahkan kombinasi kosong pun dianggap valid dan membawa keberuntungan karena memiliki panjang 0 (yang merupakan bilangan genap).

Namun, Kitty bukan tipe yang suka menghitung manual. Ia meminta bantuanmu untuk membuat program yang dapat **mencetak semua subsekuens pita** yang memiliki jumlah elemen **genap**, dan **menghitung total banyaknya kombinasi** tersebut. Selain itu, Kitty juga meminta agar keluaran dicetak dengan urutan yang memiliki **jumlah elemen subsekuens terbanyak ke paling sedikit**, dan urutan elemen di dalam setiap subsekuens tetap **mengikuti urutan indeks aslinya** (dari kiri ke kanan) agar terlihat rapi dan cantik.

Tentu saja, Kitty punya satu syarat penting, yakni kamu harus menggunakan **rekursi** untuk membangkitkan semua kombinasi. Ia tidak suka pendekatan iteratif seperti *for*, *while*, atau *do-while*.

Format Masukan

Baris pertama berisi sebuah bilangan bulat n , yakni jumlah pita yang dimiliki Hello Kitty ($1 \leq n \leq 13$).

Baris kedua berisi n bilangan bulat p_1, p_2, \dots, p_n , yakni kode angka dari masing-masing pita ($10^{-9} \leq p_i \leq 10^9$).

Format Keluaran

Cetak semua subsekuens yang memiliki **jumlah elemen genap** per baris dengan urutan dari **jumlah elemen subsekuens terbesar ke terkecil** melalui format berikut.

$$\{ a_1 a_2 \dots a_k \}$$

Jika subsekuens kosong, cetak $\{ \}$.

Setelah semua subsekuens dicetak, berikan baris kosong. Lalu, cetak satu baris akhir dengan format berikut.

“Ada X total kombinasi pita genap yang dimiliki oleh Kitty” (**tanpa tanda petik**)
di mana X adalah jumlah total subsekuens yang dicetak.

Contoh Masukan #1

```
3
5 20 30
```

Contoh Keluaran #1

```
{ 5 20 }
{ 5 30 }
{ 20 30 }
{ }
```

Ada 4 total kombinasi pita genap yang dimiliki oleh Kitty

Contoh Masukan #2

```
4
1 2 3 4
```

Contoh Keluaran #2

```
{ 1 2 3 4 }
{ 1 2 }
{ 1 3 }
{ 1 4 }
{ 2 3 }
{ 2 4 }
{ 3 4 }
{ }
```

Ada 8 total kombinasi pita genap yang dimiliki oleh Kitty

Link hint (akan diberikan sekitar 30 menit terakhir pengerjaan):

https://drive.google.com/drive/folders/17qcNVPXf01LWIZz8cs5UI9wCnUiDmyhv?usp=drive_link

Hello Kitty dan Pita Keberuntungan

By: Dimas Setiaji

| | |
|--------------|--------|
| Time Limit | 1 s |
| Memory Limit | 256 MB |

Kitty White is a collector of colorful ribbons from the village of Sanrio Puroland. Each ribbon she owns has a unique numerical code representing its color and meaning of luck.

One afternoon, Kitty sat in her collection room and began thinking about all the possible combinations of ribbons she could arrange. She wanted to try various arrangements of ribbons but was only interested in combinations with an even number of elements. According to local beliefs, even combinations bring good luck and harmony into her life.

Kitty defines a ribbon combination as a subsequence of the ribbons she owns. This means she can select some ribbons in their original order, without needing to include all of them. Even the empty combination is considered valid and lucky since it has a length of 0 (which is an even number).

However, Kitty is not the type who likes to count manually. She asks for your help to create a program that can **print all ribbon subsequences** with an **even number of elements** and **count the total number** of such combinations. In addition, Kitty also requested that the output be printed in the order of having **the largest to the fewest** number of subsequence elements, and **the order of the elements** within each subsequence still **follows the original index** order (from left to right) to make it look neat and beautiful.

Of course, Kitty has one important condition. You must use **recursion** to generate all combinations. She dislikes iterative approaches such as *for*, *while*, or *do-while*.

Input Format

The first line contains an integer n , which is the number of ribbons owned by Hello Kitty ($1 \leq n \leq 13$).

The second line contains n integers p_1, p_2, \dots, p_n , which are the numerical codes of each ribbon ($10^{-9} \leq p_i \leq 10^9$).

Output Format

CPrint all subsequences that have an **even number of elements** per line in order from the largest subsequence length to the smallest in the following format.

$$\{ a_1 a_2 \dots a_k \}$$

If the subsequence is empty, print $\{ \}$.

After all the subsequences are printed, leave a blank line. Then, print one final line with the following format.

“Ada X total kombinasi pita genap yang dimiliki oleh Kitty” **(without quotation marks)**
where X is the total number of subsequences printed.

Example Input #1

```
3
5 20 30
```

Example Output #1

```
{ 5 20 }
{ 5 30 }
{ 20 30 }
{ }
```

Ada 4 total kombinasi pita genap yang dimiliki oleh Kitty

Example Input #2

```
4
1 2 3 4
```

Example Output #2

```
{ 1 2 3 4 }
{ 1 2 }
{ 1 3 }
{ 1 4 }
{ 2 3 }
{ 2 4 }
{ 3 4 }
{ }
```

Ada 8 total kombinasi pita genap yang dimiliki oleh Kitty

Hint link (will be given approximately 30 minutes before the end of the session):

https://drive.google.com/drive/folders/17qcNVPXf01LWIZz8cs5UI9wCnUiDmyhv?usp=drive_link