

Citiri numere  $\in \mathbb{N}$  până la întâlnirea unui 0.  
Creați cu ele un vector alocat dinamic. Se  
introduc cel mult 100 de valori.

Să numărăm câte elemente dintre acestea  
sunt pătrate perfecte.

(IVU) Se vor face parcurgeri pe indici întregi.

Indicație: creăm vectorul alocând 101 elemente  
și reținem inclusiv 0-ul final.

```
int *a, *ia,
a = new int[101];
ia = a;
```

```
do
{ cin >> *ia;
  ia++;
} while( *ia != 0);
```

```
int npp = 0;
```

```
for(ia = a; *ia; ia++)
```

```
if( sqrt(*ia) == (int)sqrt(*ia))
  npp++;
```

```
cout << npp;
cout << "\n";
```

Construiți analog un vector  
de int. Afișați  
perechile neordonate de elem.  
prime între ele (cmmdc == 1)

Ex: 14, 15, 21, 8, 10, 0

răsp: 14 15

15 8

21 8

21 10

parcurgera de perechi neordonate:

```
for (ia = a; san *ia; ia++)  
    for (ib = ia + 1; *ib; ib++)  
        if (memcmp(*ia, *ib) == 1) ...
```

---

Cititi un vector analog. Afişati toate tripletle  
de vecini.

Ex:

4, 7, 9, 3, 8  
4 7 9  
7 9 3  
9 3 8