

# Cached Fibonacci

## Descriere

Una dintre cele mai simple probleme pentru recursivitate este calcularea termenului  $N$  din șirul lui Fibonacci ( $f_1 = 1, f_2 = 1$ ). Andrei dorește să facă o implementare recursivă pentru acest algoritm, dar observă că pentru valori mari ale lui  $N$ , algoritmul nu se termină rapid.

El este fan al programării funcționale, așa că nu dorește să schimbe modul recursiv de a calcula termenii din șir și alege o soluție ingenierescă pentru a rezolva problema timpului. Astfel, pentru fiecare termen anterior deja calculat, îl va reține într-un cache pentru a nu mai fi calculat în viitor.

Din păcate, el nu este familiar cu sintaxa de C++ modern, lucrând doar în limbaje pur funcționale până acum. Rezolvați bug-urile din codul lui Andrei, fără a schimba modul recursiv de calculare a termenului fibonacci.

Download sursă [aici](#).

## Formatul datelor de intrare

Pe prima linie un număr de query-uri  $T$ . Pe următoarele  $T$  linii, câte un număr natural  $p$  pentru care se dorește calcularea recursivă a termenului  $p$  din șirul lui Fibonacci.

## Formatul datelor de ieșire

Răspunsul pentru fiecare query.

## Note

Cu excepția celor deja prezente în schelet, nu aveți voie să folosiți următoarele șiruri de caractere în codul vostru: `return, exit, goto, while, for, do, fork, //, /*, /**, #define`.

## Exemple

### Exemplul 1

#### Intrare

```
3
2
3
4
```

#### Ieșire

```
1
2
3
```

