

```
#define megisto_plithos ???
```

```
??? createFibonacciSet(???, ???);
```

```
??? isFibonacci(???, ???);
```

```
main()
```

```
{
    στοιχείο_συνόλου max;
    στοιχείο_συνόλου tmp;
    Δήλωσε το σύνολο fibonacci;

    Επανάλαβε {
        Διάβασε το max
    } Μέχρι το max να ανήκει στο [2...1000]

    Κάλυψε τη συνάρτηση createFibonacciSet με παραμέτρους max και fibonacci
    Εμφάνισε το σύνολο fibonacci

    Επανάλαβε{
        Διάβασε τον αριθμό tmp
        Αν ο tmp ανήκει στο [0...1000]
            Αν ο tmp ανήκει στην ακολουθία Fibonacci (isFibonacci(???,???))
                Εμφάνισε "Fibonacci"
            Αλλιώς
                Εμφάνισε "Not Fibonacci"
        }Μέχρι ο tmp να είναι αρνητικός
    }
```

```
void createFibonacciSet(στοιχείο_συνόλου threshold, typos_συνόλου synolo){
```

```
    //Ακολουθία Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
```

```
    // Οι 2 πρώτοι αριθμοί του συνόλου είναι οι 0 και 1
```

```
    //Ο επόμενος αριθμός της ακολουθίας ορίζεται ως το άθροισμα των δύο προηγούμενων
```

```
    Δήλωσε f1, f2, tmp //όπου f1,f2 οι δύο προηγούμενοι αριθμοί στην ακολουθία
                        // και tmp βοηθητική μεταβλητή
```

```
    Δημιούργησε το σύνολο synolo
```

```
    Αρχικοποίησε f1 και f2 στους δύο πρώτους αριθμούς της ακολουθίας Fibonacci
```

```
    Εισήγαγε το f1 στο σύνολο synolo (Eisagogi(???,???))
```

```
    Επανάλαβε{
```

```
        Εισήγαγε το f2 στο σύνολο synolo (Eisagogi(???,???))
```

```
        Υπολόγισε το επόμενο στοιχείο της ακολουθίας Fibonacci
```

```
        Εκχώρησε στο f1 την παλαιά τιμή του f2
```

```
        Εκχώρησε στο f2 τη νέα τιμή που υπολόγισες
```

```
    } όσο το f2 είναι μικρότερο ή ίσο του threshold
```

```
}
```

```
boolean isFibonacci(στοιχείο_συνόλου candidate, typos_συνόλου synoloFibonacci)
```

```
{
```

```
    Κάλυψε τη συνάρτηση Melos για να ελέγξεις αν το candidate ανήκει στο σύνολο synoloFibonacci
    (Melos(???,???)
```

```
    Επέστρεψε την αντίστοιχη τιμή
```

```
}
```