```
f ( a, ... ) {

        return          ;
}


        int
int   f ( u ) {
        if  ( u  ==  0 )
                return  0;
        else if ( u   >   0 )
                return  f(u - 1);
        else
                return  f(u + v + 5;
}
```

fibonacci sequence

$$1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

```
fib (n) {                    عدد n داخل فيبوناتشي

    if   (n == 1  or  n == 2)
            return 1;
    return   f(n-1) + f(n-2);
}
```

```
boolean is_palindrome (s) {   اذا كانت بالستيرنج S[
    if ( s.equals (""))
        return True;
    if( s.length() == 1)
        return True;
    if (s. charAt (@) == s.charAt (-1))
        return  is_palindrome (s.substring(1, -1));
    return  false;
}
```
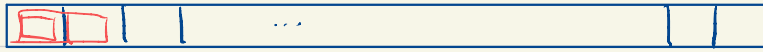
به چند طریق می‌توانیم ✓ ستونی به طول $1 \times n$ را با کاشی‌های ۱×۱ و ۲×۱ بپوشانیم؟

$f(n)$

$$f(n-1) + f(n-2)$$

$f(1) = 1$

$f(2) = 2$

$$\forall i . \ u_i \geq 0$$

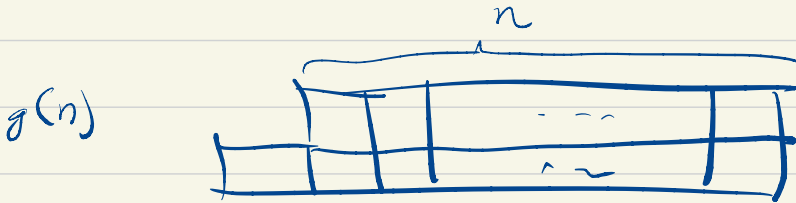$$u_1 + u_2 + \cdots + u_k = S$$

$f(k, s)$

$$\sum_{i=0}^{s} f(k-1, s-i)$$

$$f(1, n) = 1$$

$$2 \quad \underset{n}{\boxed{}} \qquad \square \quad \boxed{\square}$$

به چند طریق میتوان یک ۲×n جدول را با کاشی‌های ۱×۱ و ۱×۲ پوشاند؟
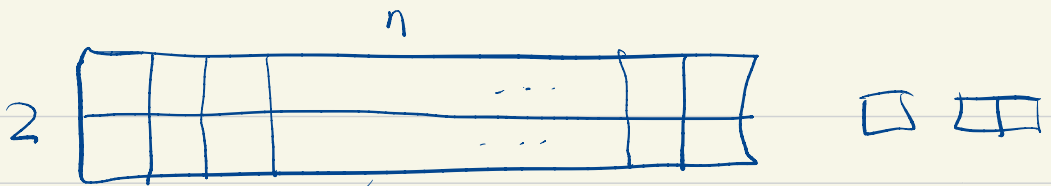
$$\begin{cases} \boxplus & f(n-1) \\ \square & g(n-1) \\ \boxed{\square\square} \end{cases} \longrightarrow \begin{cases} \boxed{\square\square} & f(n-2) \\ \square & g(n-2) \end{cases}$$

$$f(n) = f(n-1) + g(n-1) + f(n-2) + g(n-2)$$

$g(n)$

$$\begin{cases} \boxed{\square} & f(n) \\ \boxed{\square\square} & g(n-1) \end{cases}$$

$$g(n) = f(n) + g(n-1)$$

$$g(0) = 1 \qquad f(1) = 2$$
$$g(1) = 3 \qquad f(2) = 7$$

1000, 1, 5, 4, 7, 6, 2, 10

array f(array
a) { ... آخر , a مرتب شده بحالتی که بکنید.

```
n :=   |a|
l :=   نصف اول a
r :=   نیم دوم a
ls := f(l)
rs := f(r)
ans := {}
i := 0
j := 0
while (ans.size() < n) {
        if (ls[i] < rs[j])
                ans.add(ls[i]);
                i++;
        else
                ans.add(rs[j]);
                j++;
        }
    return ans;
}
```

$$S = \{ \qquad \}$$

به طور اُتمایل همهٔ کی زیر مجموعهای S را پایدا پیپیم.

f(S) {   همهٔ کی زیر مجموعهای S

```
if (|S| == 0)
    return { {} };
n := یک عضو دلخواه از S
T = S \ {n}
A = f(T)
B = {}
for (y in A) {
    B.add (y);
    B.add (y ∪ {n});
}
}
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$