

Ejercicio 1: Calcular funciones matemáticas

Un compañero tuyo vio las cápsulas de OCILabs y aprendió que podía calcular funciones matemáticas utilizando algoritmos recursivos. Pero, él no entendió muy bien las cápsulas, entonces te pide ayuda para programar varios programas relacionados con eso.

Primera pregunta: Fibonacci

Tu compañero quiere primero aprender a programar la función de Fibonacci que vio en las cápsulas. Recordar que la función de Fibonacci se escribe de la siguiente forma $F(n)$ y dependiendo del valor de n retornará el siguiente valor:

- Si $n = 1$, entonces $F(n) = 1$
- Si $n = 2$, entonces $F(n) = 2$
- Si $n > 2$, entonces $F(n) = F(n - 1) + F(n - 2)$

Para eso le pide hacer un programa que lea un entero n y utilizando una función recursiva muestre por pantalla el resultado de Fibonacci

Input de ejemplo

1 4

Output de ejemplo

1 3

Input de ejemplo

1 5

Output de ejemplo

1 5

Input de ejemplo

1 6

Output de ejemplo

1 8

Segunda pregunta: Otra función matemática

Tu compañero ahora se pregunta si se podría hacer lo mismo con otro tipo de funciones matemáticas, en este caso él busca sobre una misteriosa sucesión llamada: la sucesión de Mario, la cual denotaremos $M(n)$.

La sucesión de Mario recibe un entero n y tiene el siguiente comportamiento:

- Si $n = 1$, entonces $M(n) = 1$.
- Si $n = 2$, entonces $M(n) = 2$.
- Si n es un número par, entonces $M(n) = M(n/2) + M(n/3)$

- Si n es un número impar, entonces $M(n) = M(n - 1) + M(n - 2)$

Para ver los resultados de esta función matemática, le pide hacer un programa que lea un entero n y utilizando una función recursiva muestre por pantalla el resultado de la sucesión de Mario.

Input de ejemplo

```
1 4
```

Output de ejemplo

```
1 3
```

Input de ejemplo

```
1 5
```

Output de ejemplo

```
1 6
```

Input de ejemplo

```
1 6
```

Output de ejemplo

```
1 5
```

Ejercicio 2: Funciones recursivas en otros lados

Tu compañero se da cuenta que en se pueden hacer funciones recursivas en otros lados.

Primera pregunta: Palíndromos

Los palíndromos son palabras que se escriben igual de izquierda a derecha y de derecha a izquierda, por ejemplo *alela*. Tu compañero se da cuenta que verificar si un palabra es palíndromo o no se puede realizar una función recursiva para eso.

Tu compañero dice que:

- Si la palabra es de largo 1, entonces siempre es palíndromo
- En cualquier otro caso, se retorna el siguiente valor:

```
1 palabra[pos_inicial] == palabra[pos_final] && palindromo(palabra,
    pos_inicial + 1, pos_final - 1);
```

Para esto, tú sabes que la función recursiva debe recibir 3 parámetros la palabra, la posición inicial y la final.

```
1 bool palindromo(string palabra, int inicio, int final);
```

Escriba un programa que reciba un string correspondiente a una palabra y muestra por pantalla *ES PALINDROMO!!* si la palabra es palíndromo, sino muestra *NO ES PALINDROMO :C*.

Input de ejemplo

```
1 a
```

Output de ejemplo

```
1 ES PALINDROMO!!
```

Input de ejemplo

```
1 aasdlkasjda
```

Output de ejemplo

```
1 NO ES PALINDROMO :C
```

Input de ejemplo

```
1 alabarabala
```

Output de ejemplo

```
1 ES PALINDROMO!!
```

Segunda pregunta: dos números sumados dan K

Ahora, su compañero les quiere dar un desafío. Él les dice: “Quiero que escribas una función recursiva que reciba un vector de enteros y pueda determinar si es que dos números sumados dan K o no”.

Y tú, valientemente le respondes: “DESAFÍO ACEPTADO!”.

Entonces, tú debes escribir un programar que lea un vector de enteros y un entero K , y muestre por pantalla 1 si es que dos números del vector sumados dan como resultado K , sino muestre 0.

La primera línea contendrá dos entero n y k , correspondiente a la cantidad de números en el vector y el valor de k respectivamente.

La segunda línea contendrá n enteros separados por un espacio.

Input de ejemplo

```
1 4 2
2 1 4 3 2
```

Output de ejemplo

```
1 0
```

Input de ejemplo

```
1 4 3
2 1 4 3 2
```

Output de ejemplo

```
1 1
```