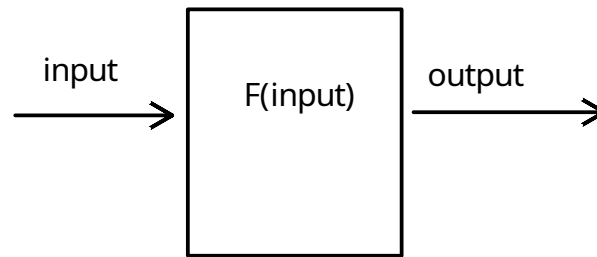
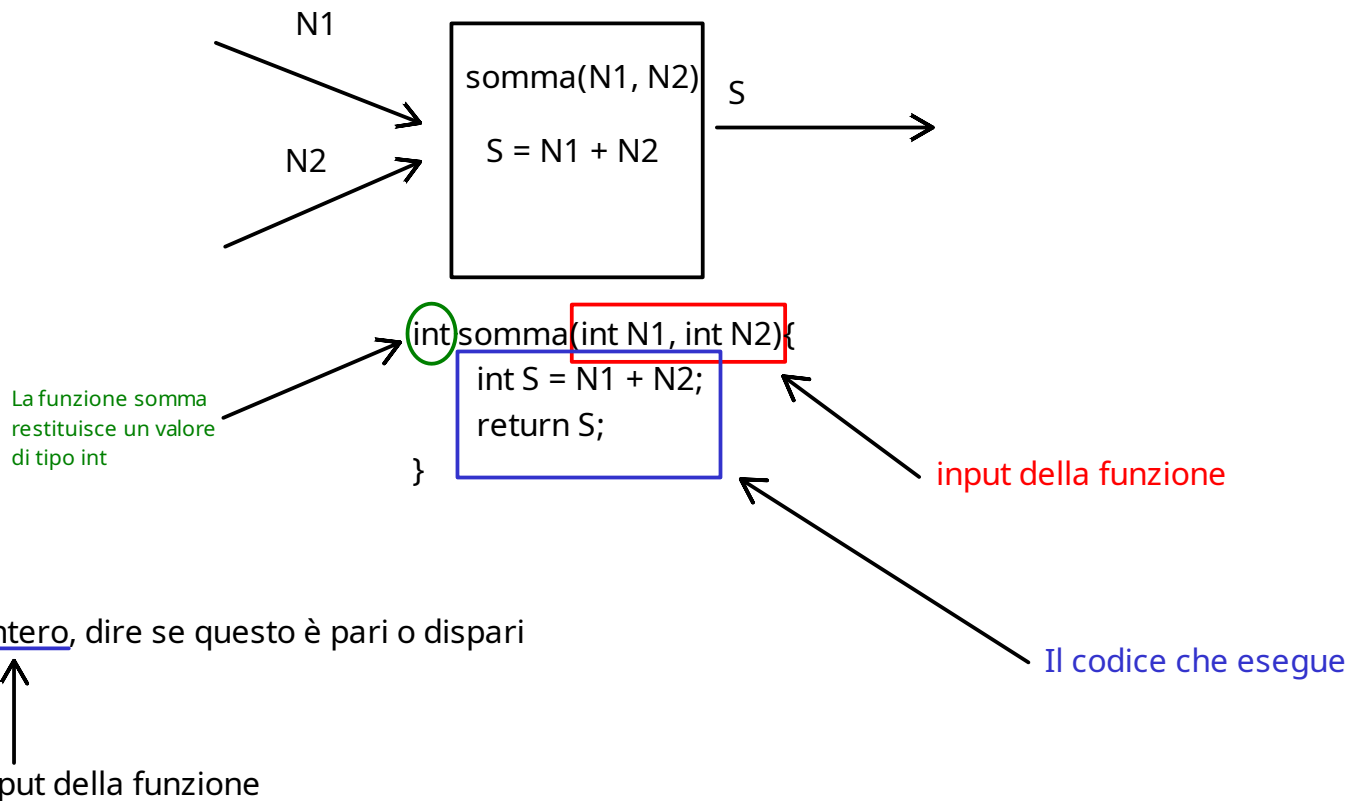


printf(); Funzioni "black box (scatola nera)", non sappiamo come sono fatte all'interno, ma  
 scanf(); sappiamo cosa dovrebbero fare.  
 rand();

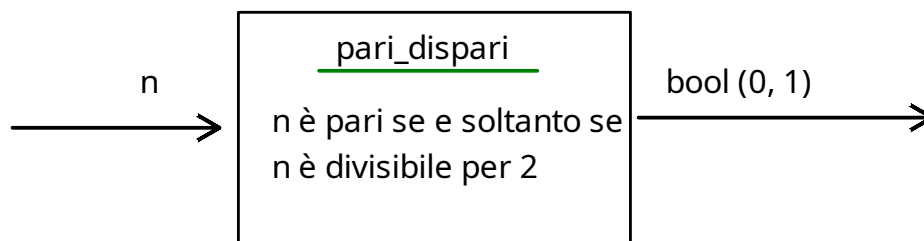


Obiettivo: Imparare a scrivere funzioni proprie!!!



Dato un intero, dire se questo è pari o dispari

↑  
l' input della funzione



- Se il numero in input è pari, la funzione deve restituire TRUE, ovvero 1
- Se il numero in input è dispari, la funzione deve restituire FALSE, ovvero 0

```

int pari_dispari(int n){
    int resto = n % 2; // % è mod
    // n % 2: resto di n / 2
    if(resto == 0) // n è pari
        return 1;
    return 0;
}
  
```

6 / 2 = 3 => 6 / 2 = 3 con resto di 0 => 6 è pari  
 7 / 2 = 3.5 => 7 / 2 = 3 con resto di 1 => 7 è dispari

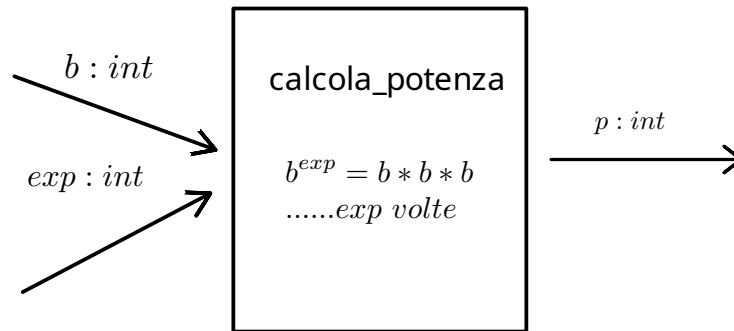
In C, per trovare il resto della divisione, si usa l'operatore %

6 % 2 = 0 => 6 è pari  
 7 % 2 = 1 => 7 è dispari

Un numero è divisibile per 2, se il resto della divisione è 0

Dati due numeri interi, "base (b)" e "esponente (exp)", calcolare  $b^{exp}$

Esempio:  $b = 2, exp = 3 \Rightarrow 2^3 = 2 * 2 * 2 = (2 * 2) * 2 = 4 * 2 = 8$



```
int calcola_potenza(int b, int exp){  
    // Codice che calcola  $b^{exp}$   
    int i, potenza = 1;  
    for (i = 0; i < exp; i++){  
        potenza = potenza * b;  
    }  
    return potenza;  
}
```

Per casa: Implementare queste funzioni in C, e fare delle prove, chiamandole nel main

Calcolare se questi numeri sono pari o dispari: `int numeri[] = {2, 3, 5, 10}`  
le seguenti potenze: `{2^3, 5^10}`

## RIEPILOGO:

### DEF (Funzione):

Una funzione è un blocco di codice, che viene eseguito quando viene chiamata la funzione.

Una funzione è costituita dalle seguenti caratteristiche: Input, Nome, Blocco di codice, Output.

In C una funzione si dichiara nel seguente modo:

```
Tipo_Dato_Ritorno Nome_Funzione(Tipo Input1, Tipo Input2, ..., Tipo Input n){  
    Blocco di codice  
    Se la funzione ritorna un valore, allora return "valore"  
}
```