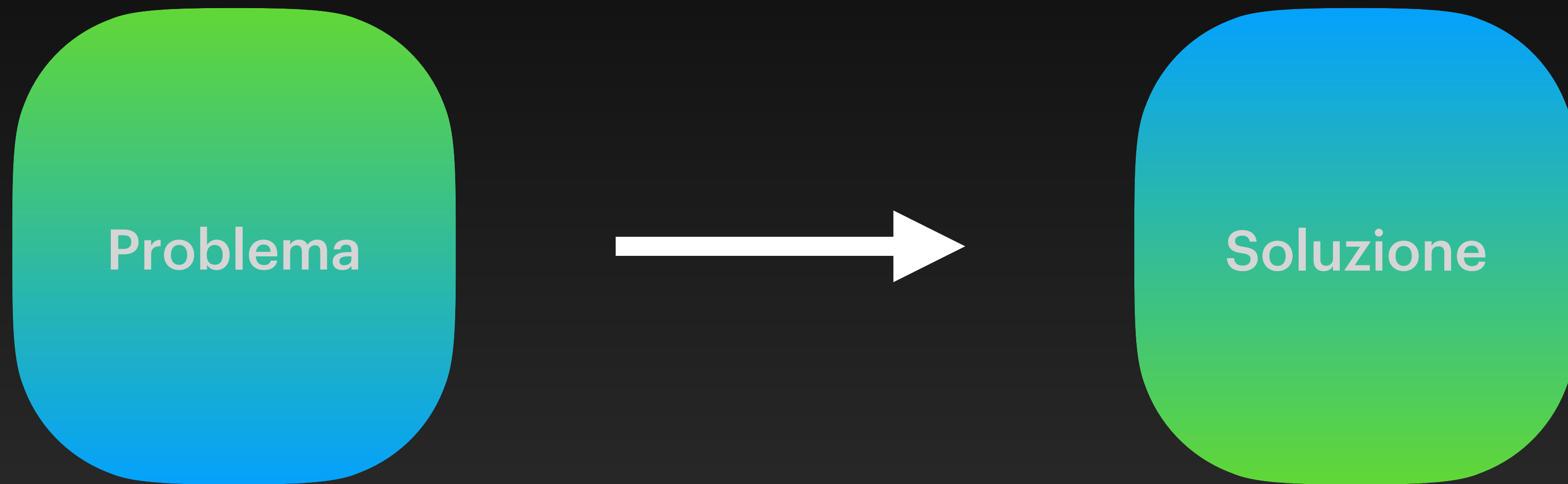


Programmare

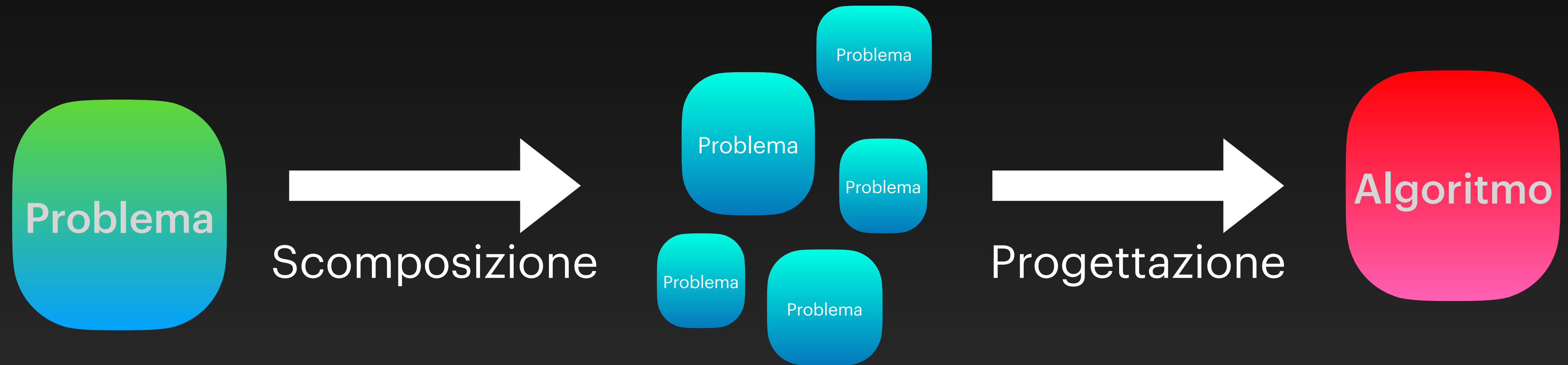
Come smettere di essere spettatore

Emanuele Moro

Cosa significa creare software?



Cosa significa creare software?

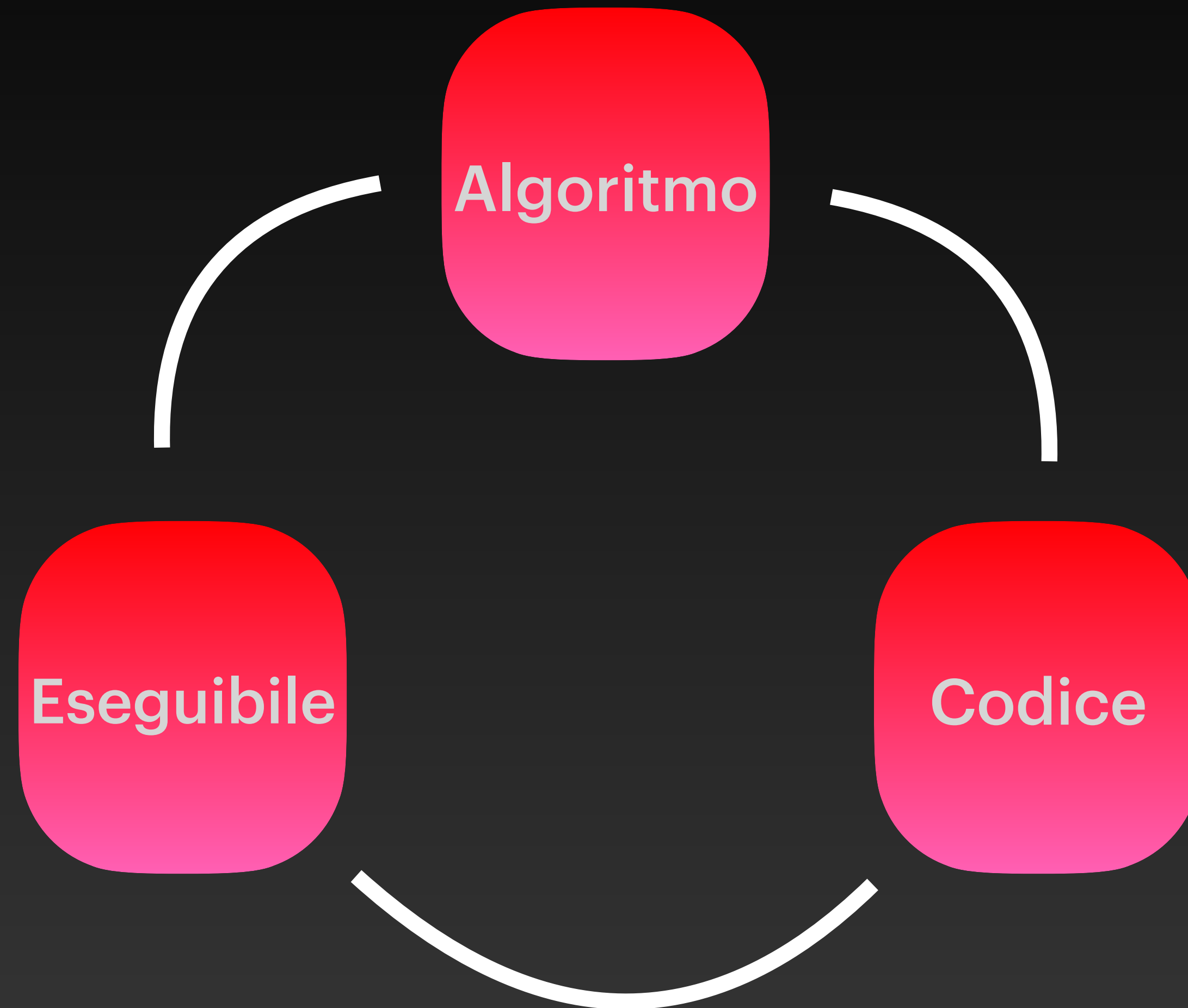


Cosa significa creare software?



Cosa significa creare software?

Debugging



Concetti di base

“Implementare un algoritmo” significa tradurre una serie di operazioni aritmetico-logiche in procedure eseguibili dal calcolatore

Professor Pier Luca Montessoro

Algoritmo

Una sequenza **finita** di istruzioni che descrivono **eseguire** un compito in modo **non ambiguo**.
Ogni passaggio è ben definito, deve essere eseguito in un ordine preciso e porta a un risultato specifico.

Variabile

Una scatola dove puoi inserire un valore (dato), ma anche leggerlo e modificarlo

Funzione

Una porzione di codice, che può avere dei parametri in input, ha un comportamento e può avere un valore di restituzione

L'importanza delle funzioni nella struttura del codice

Differenza tra spaghetti code e procedurale

1. Inserisci primo valore
2. Inserisci secondo valore
3. Inserisci operazione
4. Se operazione è somma vai a 8
5. Se operazione è prodotto vai a 13
6. Se operazione è sottrazione vai a 14
7. Se operazione è quoziente vai a 15
8. Restituisci la somma e vai a 9
9. Chiedi se vuoi continuare
10. Se risposta è Si vai a 1
11. Se risposta è No vai a 12
12. Exit
13. Restituisci il prodotto e vai a 9
14. Restituisci la sottrazione e vai a 9
15. Restituisci il quoziente e vai a 9

1. Func somma {...}
2. Func sottrazione {...}
3. Func quoziente {...}
4. Func prodotto {...}
5. Func calcolatrice {...}
6. Func main {
 Inserisci a
 Inserisci b
 Inserisci operatore
 Ris = Calcolatrice(a, b, operatore)
 // chiedi se vuole continuare
}

Ciaooooooooo mondo!

Come è strutturato un programma in C?

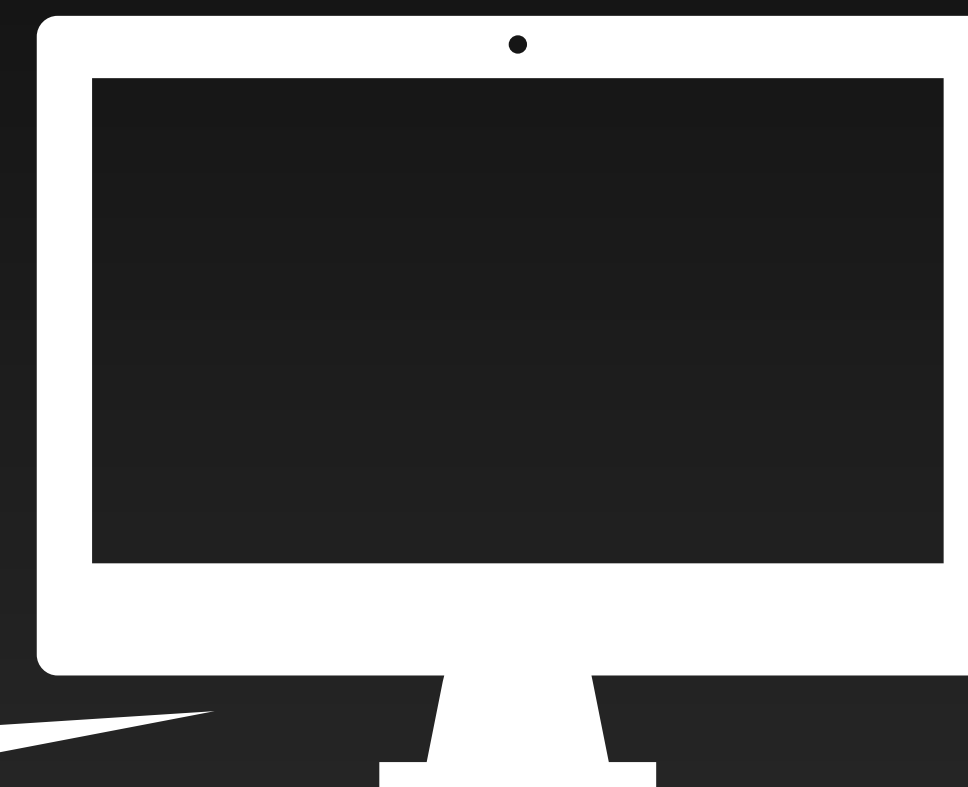
- Alle righe 1 e 2 troviamo delle **direttive**:
 - Si riconoscono dal hashtag
 - In questo caso le usiamo per includere delle librerie
- A riga 5 c'è la **funzione main**, dentro le graffe si trova il corpo della funzione
- A riga 4 e 10 troviamo dei **commenti**, vengono ignorati dal compilatore

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* programma principale */
5  int main()
6  {
7      printf("Hello world!\n");
8      return EXIT_SUCCESS;
9  }
10 // fine del programma
11
12
13
14
15
16
17
18
```

Una questione di linguaggio

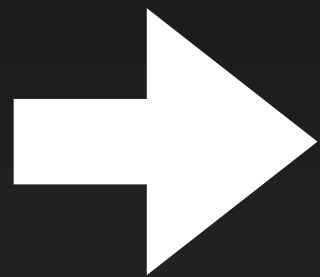
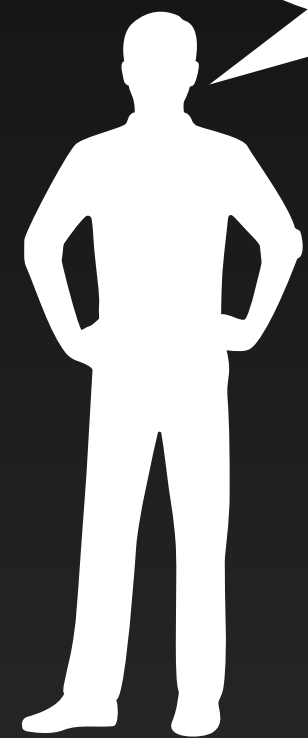


01001001000100
01000100100100
10010010010000

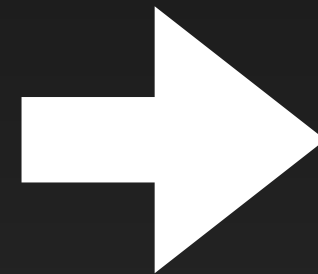


Una questione di linguaggio

Calcolami...



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* programma principale */
5  int main()
6  {
7      printf("Hello world!\n");
8      return EXIT_SUCCESS;
9  }
10 // fine del programma
11
12
13
14
15
16
17
18
```

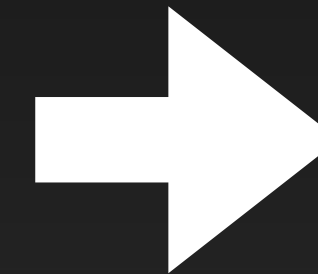


```
section .text
global _start                                ;must be declared for linker (ld)
_start:                                       ;tell linker entry point

    mov     edx,len                          ;message length
    mov     ecx,msg                         ;message to write
    mov     ebx,1                           ;file descriptor (stdout)
    mov     eax,4                           ;system call number (sys_write)
    int     0x80                             ;call kernel

    mov     eax,1                           ;system call number (sys_exit)
    int     0x80                             ;call kernel

section .data
msg     db 'Hello, world!',0xa              ;our dear string
len     equ $ - msg                        ;length of our dear string
```



Una questione di linguaggio

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* programma principale */
5  int main()
6  {
7      printf("Hello world!\n");
8      return EXIT_SUCCESS;
9  }
10 // fine del programma
11
12
13
14
15
16
17
18
```



Compilatore e Linker

```
section      .text
global      _start

_start:

    mov     edx,len
    mov     ecx,msg
    mov     ebx,1
    mov     eax,4
    int     0x80

    mov     eax,1
    int     0x80

section      .data

msg         db    'Hello, world!',0xa
len         equ   $ - msg
```

Compilatore

Traduce il codice sorgente scritto in C (o in altri linguaggi compilati) in codice **oggetto** (linguaggio macchina non completo)

Linker

Unisce il prodotto del compilatore, con le librerie necessarie a completare il programma eseguibile

Tutto il software è compilato?

No, altrimenti il web e il machine learning non potrebbero esistere

Interprete

Non è un compilatore, quindi non restituisce un file eseguibile, ma interpreta a tempo di esecuzione (runtime)

Una questione di linguaggio

Cos'è un linguaggio di programmazione?

- È una lingua a sé stante:
 - Ha un **lessico** limitato (molto più limitato rispetto ad una lingua umana);
 - Ha una **sintassi**;
 - Proposizioni diverse possono mantenere la stessa **semantica**.

Programmazione strutturata

Diagramma a blocchi

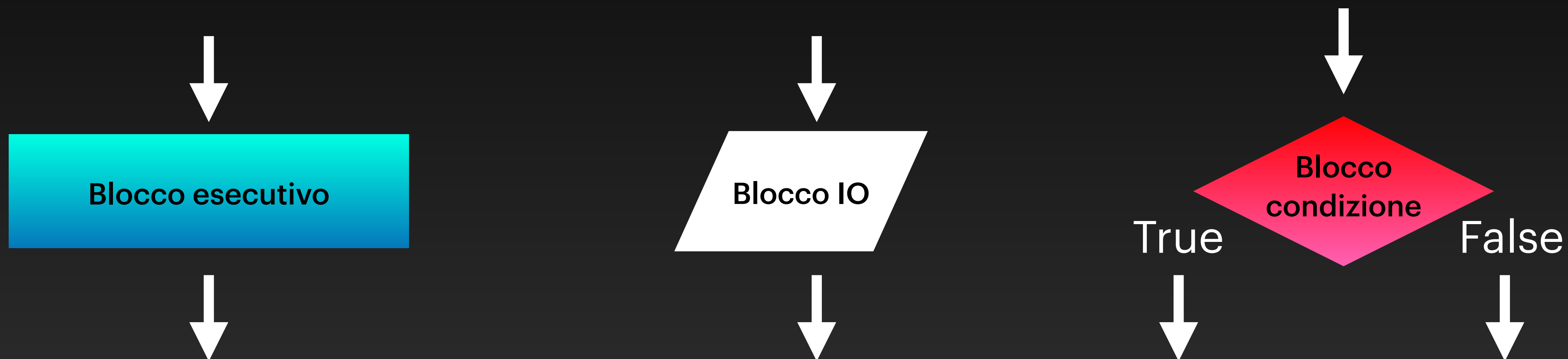


Diagramma a blocchi

Semplificazione del prof

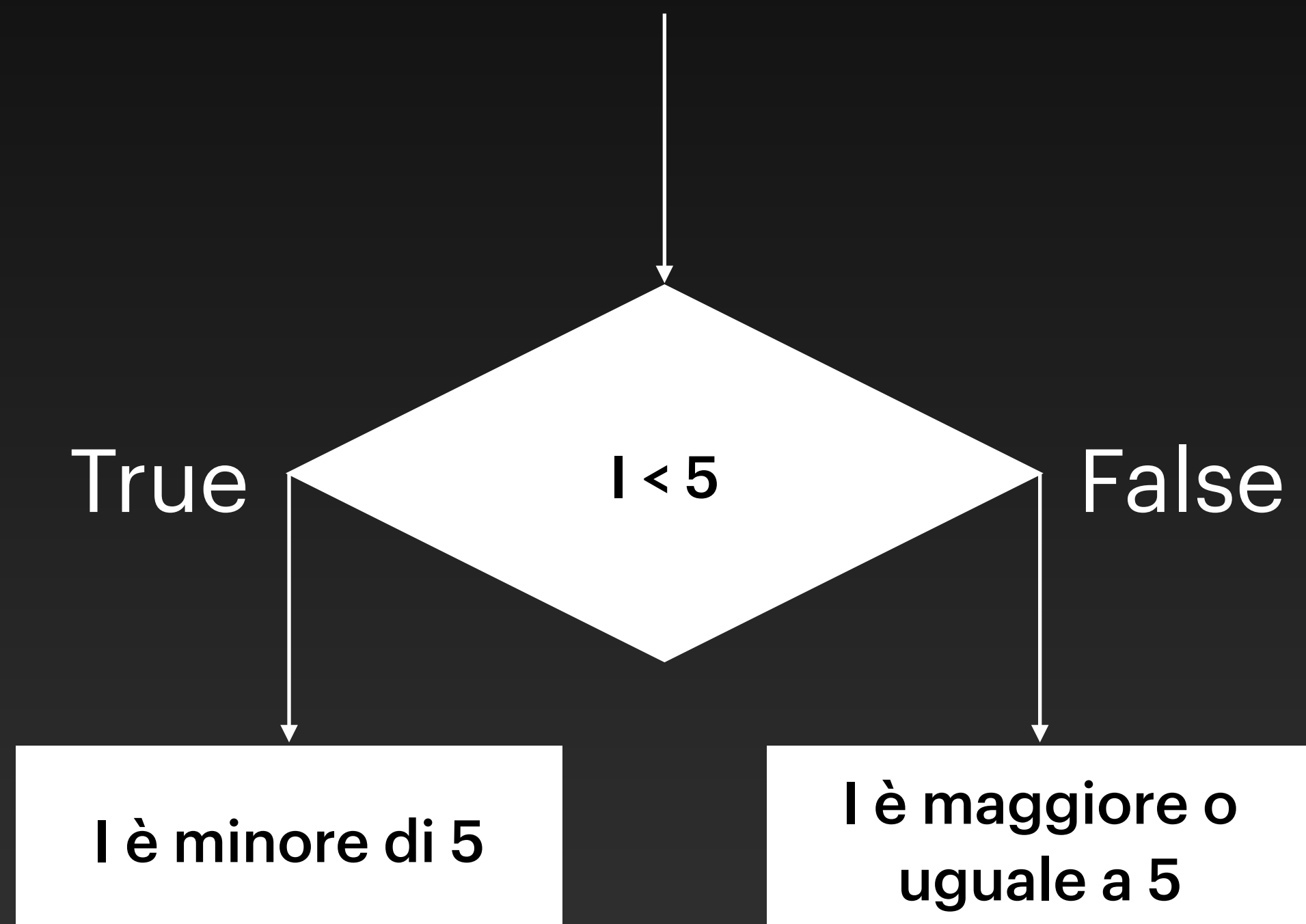


**“Ciò che si deve allenare è il processo logico
mediante il quale dalla lettura del problema si
riesce a trovare una soluzione componibile con
costrutti, non la scrittura del programma”**

Emanuele Moro

Dai blocchi ai costrutti del linguaggio C

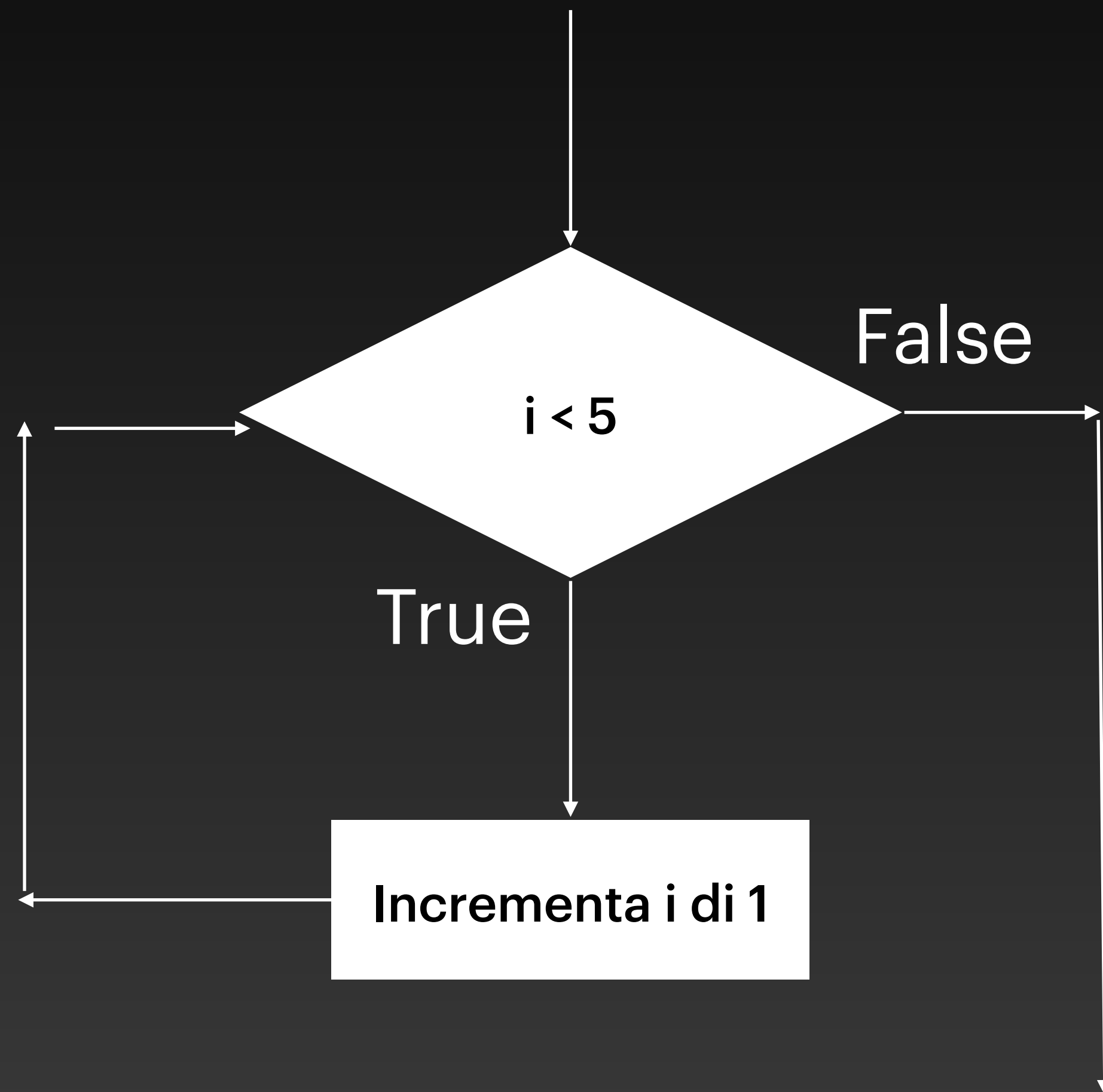
If-then-else



```
if (i < 5) {  
    printf("i è minore di 5\n");  
} else {  
    printf("i è maggiore o uguale a 5\n");  
}
```

Dai blocchi ai costrutti del linguaggio C

While



```
while(i < 5) {  
    i++;  
}
```

Prova tu

- Crea il diagramma a blocchi per calcolare $y = x^n$