UNIVERSITÀ
DI TORINO

Laboratorio di Programmazione I

Lezione n. 4: Iterazioni e correttezza

Alessandro Mazzei

Slides: prof. Elvio Amparore

Outline

- for e while: iterazioni_n.c
- Media sequenza
- CodeRunner: Lab04-Es1 Divisori Interi
- Triangolo di Floyd
- Quadrato di *
- Quantificazione Universale e Esitenziale in C
- CodeRunner: Lab04-Es2 Esercizi su condizioni
- Un esercizio sulla Correttezza
- Fibonacci
- fai_indovinare_numero.c
- Extra: Lab04-Es3 Indovina tu il numero!

Iterazioni con il for

Quando usare il for:

- quando il numero di iterazioni è noto sin dall'inizio. Tipicamente i cicli for avranno una forma del tipo:

```
for (int i=0; i<n; i++) {
        // codice che usa ma non cambia i
        ...
}
```

oppure:

```
for (int i=n; i>=0; i--) {
        // codice che usa ma non cambia i
        ...
}
```

Iterazioni con il while

Quando usare il while:

- quando la condizione di terminazione si determina **solo durante le iterazioni stesse**.
- quando non serve inizializzare una variabile di iterazione

```
bool continua = true;
while (continua) {
   ...
      if (condizione)
      continua = false;
}
```

```
int i = 0;
...
while (i < n) {
   ...
      i = i + k;
}
```

Esercizi iterativi per iniziare

Scrivere un programma **iterazioni_n.c** che implementa i seguenti punti:

1. Dato un intero **n** letto da input con scanf, si stampino a video tutti gli interi da **0** a **n** e poi da **n** a **0**.
   DOMANDA: Quale ciclo è più appropriato?
2. Successivamente si stampino a video tutti gli interi dispari da **0** a **n**
3. Infine si stampi a video **n!**, il fattoriale di **n**.

Si ricorda che il fattoriale di **n** è pari a: 1 * 2 * … * (**n**-1) * **n**.

Media di una sequenza

Scrivere un programma **media_seq.c** che:

- Legge dall'input una sequenza di numeri interi, terminata dal numero **0** (che assumiamo non appartenga alla sequenza letta).
- Quando l'utente inserisce il numero **0**, il programma interrompe la lettura della sequenza, e stampa la media (come numero intero) di tutti i numeri letti.
- Provare il programma sulle seguenti sequenze:

4 8 10 2 0 **3 5 0** **0**

SUGGERIMENTO: ci servirà un accumulatore che somma i numeri letti, e un contatore dei numeri letti, per cui:

media = accumulatore_somma / contatore

Divisori Interi

Sulla pagina Moodle trovate un esercizio con nome

Lab04-Es1 Divisori Interi

Completate il programma, assicurandovi che passi tutti i test proposti.

Quale ciclo è più adatto per il problema proposto?

Numeri di Floyd

Scrivere un programma **floyd.c** che prende in input un numero **n** e stampa le prime **n** righe del *triangolo di Floyd*.

Il triangolo di Floyd è costituito dai numeri naturali scritti in modo consecutivo, per riempire le righe con 1,2,3,... valori.
Ad esempio per **n**=5 il programma deve stampare:

```
1
2  3
4  5  6
7  8  9  10
11  12  13  14  15
```

NOTA: I numeri di Floyd non c'entrano nulla con la logica di Floyd!!

Quadrato

Scrivere un programma **quadrato.c** che legge in input un numero **n** e stampa a video un quadrato di **n*n** caratteri tra '*', '\' e ':' seguendo questo pattern (esempio per **n**=6):

```
\:::::
*\::::
**\:::
***\::
****\:
*****\
```

Suggerimento: fare due cicli annidati, uno esterno per le righe, ed uno o più cicli interni per i caratteri sulla stessa una riga.

Quantificazione universale

Supponiamo che stiamo scrivendo un programma con un ciclo che legge una sequenza di interi, terminata dallo **0**.

Vogliamo rispondere a questa domanda:

Tutti i numeri della sequenza
rispettano una condizione *Cond*?

Come possiamo fare?

```
bool tutti = true;
while (scanf("%d", &n) && n!=0) {
    if (! Cond valutata su n )
        tutti = false;
}
```

Quantificazione esistenziale

Supponiamo che stiamo scrivendo un programma con un ciclo che legge una sequenza di interi, terminata dallo **0**.

Vogliamo rispondere a questa domanda:

Esiste almeno un numero della sequenza
che rispetta una condizione *Cond*?

Come possiamo fare?

```
bool esiste = false;
while (scanf("%d", &n) && n!=0) {
    if (Cond valutata su n)
        esiste = true;
}
```

Cicli con accumulatori booleani

- Se invece non dobbiamo leggere tutti gli elementi della sequenza (ad esempio perché sono già in memoria), possiamo aggiungere la variabile sentinella `tutti (esiste)` che accumula la condizione *Cond* anche nella condizione del while.
- In questo modo il ciclo termina non appena la variabile sentinella che codifica la condizione booleana diventa falsa (universale) o vera (esistenziale).
- Vedremo in un prossimo laboratorio questo schema.

Esercizi su condizioni

Sulla pagina Moodle trovate un esercizio con nome

Lab04-Es2 Esercizi su condizioni

Completate il programma, determinando correttamente tutte le condizioni esiste/per-ogni che vengono richieste.

Correttezza - un breve ripasso

```
int int main(void) {
        // PRE-COND: 0<=abs(a) && abs(a)<=9
        int a = ...;

        // pre: ??
        //  ??
        int b = a * a;

        // pre: ??
        //  ??
        int c = b / 2;

        // POST-COND:    0<=c && c<=40
        printf("%d\n", c);
}
```

Svolgimento

```c
int int main(void) {
        // PRE-COND: 0<=abs(a) && abs(a)<=9
        int a = ...;

        // pre: ??
        //  ??
        int b = a * a;

        // pre: 0<=b/2 && b/2<=40
        //  ??
        int c = b / 2;

        // POST-COND:   0<=c && c<=40
        printf("%d\n", c);
}
```

Svolgimento

```
int int main(void) {
        // PRE-COND: 0<=abs(a) && abs(a)<=9
        int a = ...;

        // pre: ??
        //  ??
        int b = a * a;

        // pre: 0<=b/2 && b/2<=40
        //  imply 0<=b && b<=81
        int c = b / 2;

        // POST-COND:   0<=c && c<=40
        printf("%d\n", c);
}
```

Svolgimento

```
int int main(void) {
        // PRE-COND: 0<=abs(a) && abs(a)<=9
        int a = ...;

        // pre: 0<=a*a && a*a<=81
        //  ??
        int b = a * a;

        // pre: 0<=b/2 && b/2<=40
        //  imply 0<=b && b<=81
        int c = b / 2;

        // POST-COND:   0<=c && c<=40
        printf("%d\n", c);
}
```

Svolgimento

```
int int main(void) {
        // PRE-COND: 0<=abs(a) && abs(a)<=9
        int a = ...;

        // pre: 0<=a*a && a*a<=81
        //  imply 0<=abs(a) && abs(a)<=9
        int b = a * a;

        // pre: 0<=b/2 && b/2<=40
        //  imply 0<=b && b<=81
        int c = b / 2;

        // POST-COND:   0<=c && c<=40
        printf("%d\n", c);
}
```

Fibonacci

Scrivere un programma **fibonacci.c** che legge in input un numero **k** e stampa a video i primi **k** numeri della successione di Fibonacci. Ad, esempio la stampa attesa per **k = 10** è:

```
0  1  1  2  3  5  8  13  21  34
```

Per semplicità assumere che **k** >= 2.

Si ricorda che la successione di Fibonacci parte da due numeri 0 e 1, ed ogni elemento successivo della successione è ottenuto come somma dei due elementi precedenti.

Suggerimento: partite da due numeri **n=0** ed **m=1** e stampateli. Ad ogni iterazione si procede ad aggiornare le variabili:

$$n' = m, \qquad m' = n + m$$

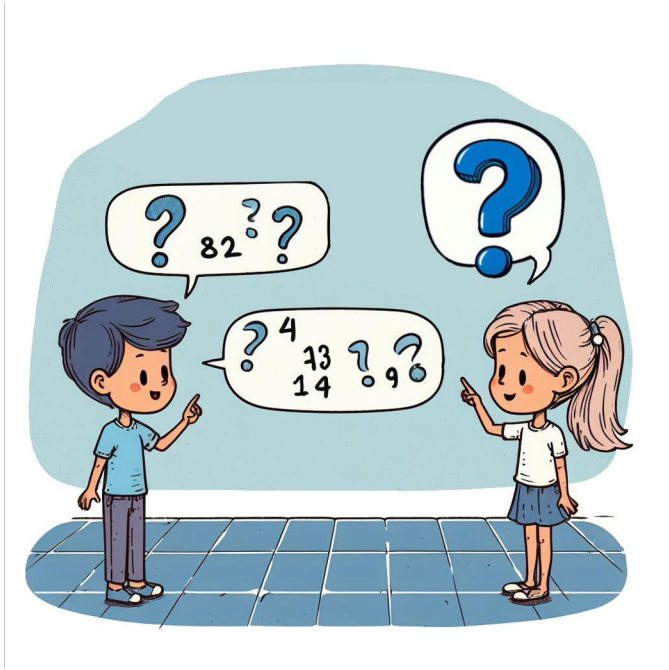
e a stampare **m'**.

Domanda: Per fare questo aggiornamento serve una variabile temporanea?

Fai indovinare un numero!

Partendo dal codice iniziale di **fai_indovinare_numero.c**, leggere con attenzione il codice e la richiesta, ed implementare il gioco.

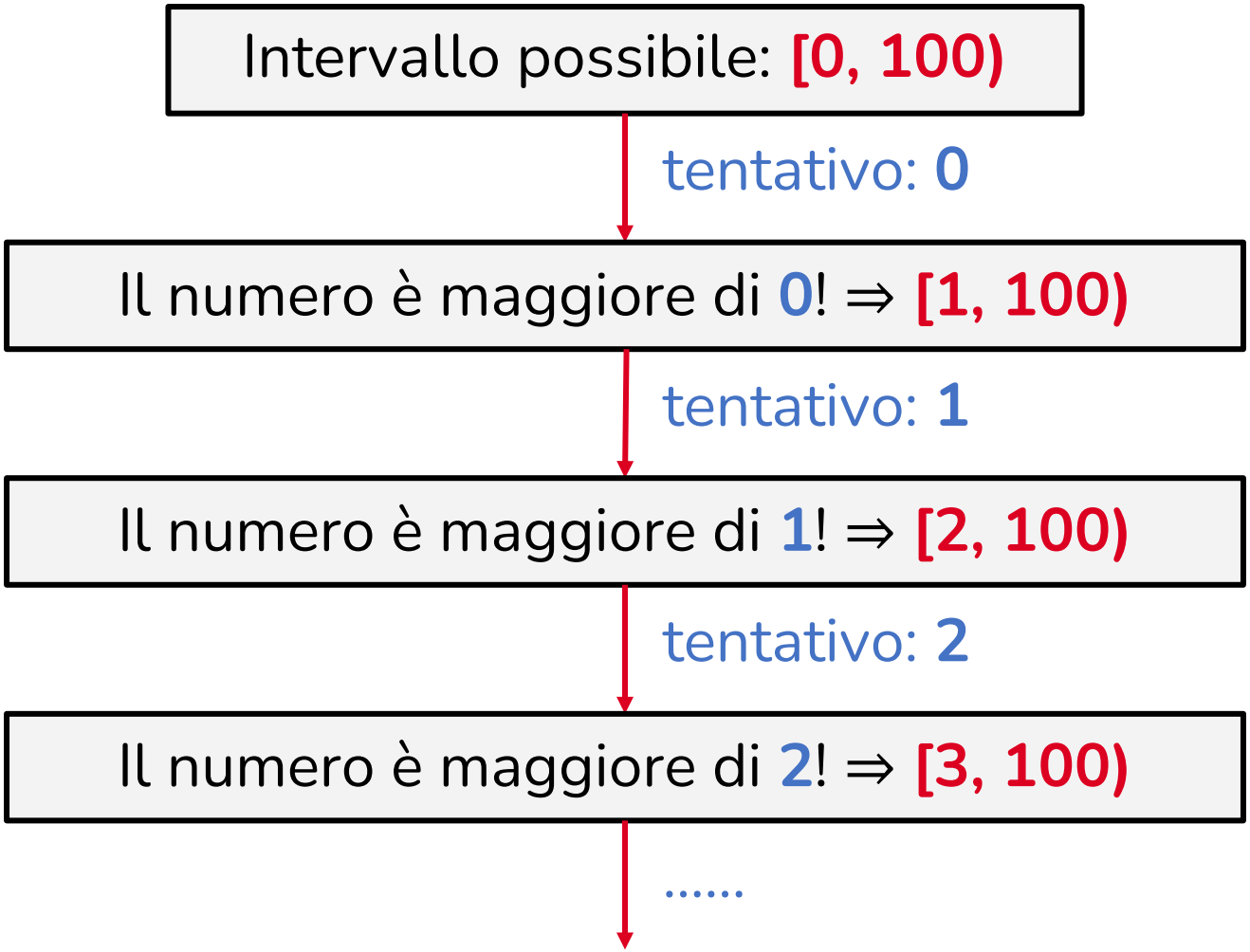
Quale strategia usa una persona per indovinare il numero casuale scelto dal programma C?



Strategia di gioco con iterazioni

ESERCIZIO EXTRA

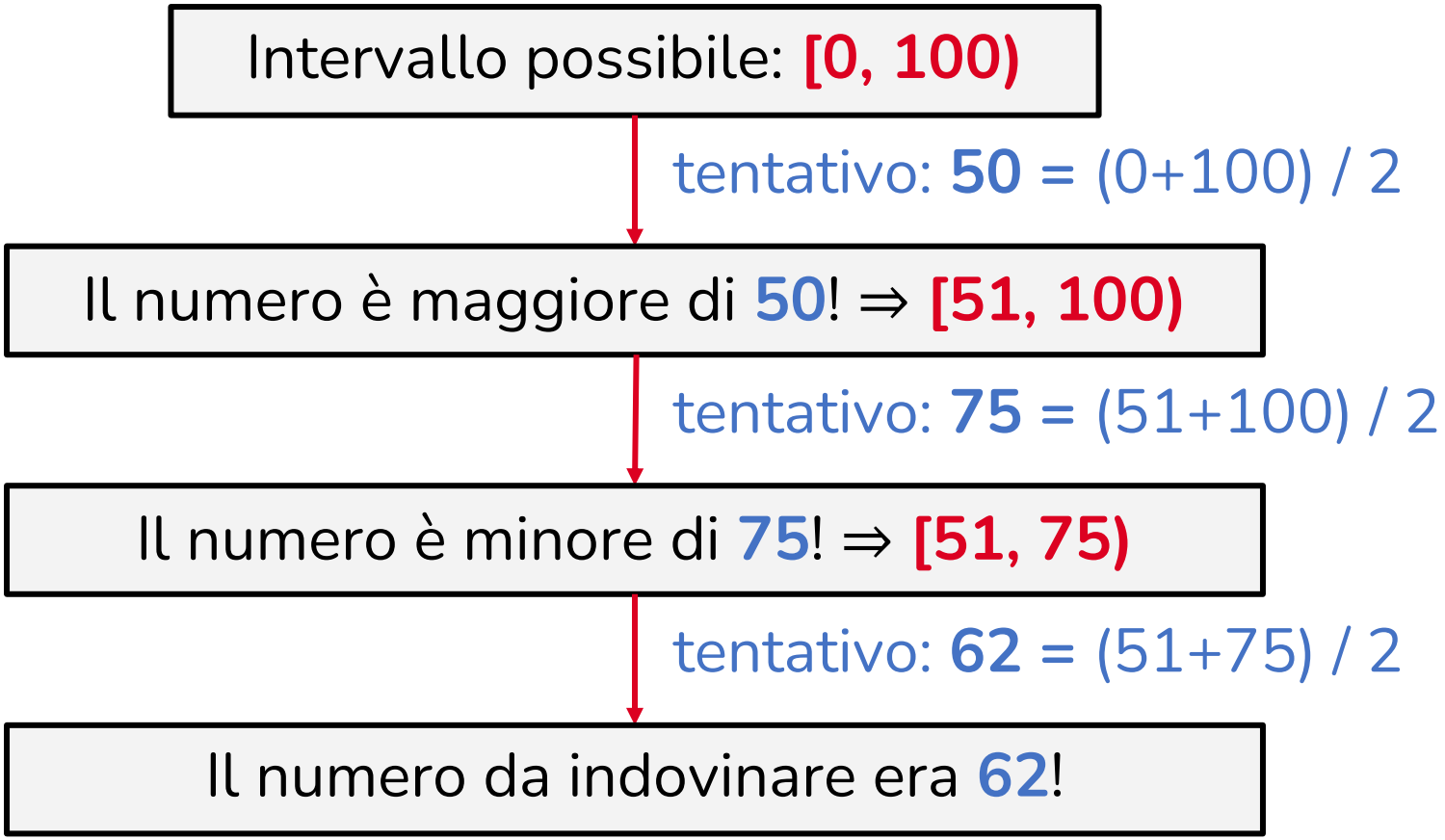
Sappiamo che il numero da indovinare è compreso tra **0** incluso e **100** escluso. Supponiamo sia **62**. Quindi:



Strategia di gioco con dimezzamenti

ESERCIZIO EXTRA

Sappiamo che il numero da indovinare è compreso tra **0** incluso e **100** escluso. Supponiamo sia **62**. Quindi:



Indovina tu il numero!

ESERCIZIO EXTRA

Adesso scrivete l'algoritmo che indovina il numero casuale che ha scelto il computer!

Sulla pagina Moodle trovate un esercizio con nome

Lab04-Es3 Indovina tu il numero!

Completate il programma, assicurandovi che indovini tutti i valori nel numero massimo di tentativi concessi.

Quale ciclo è più adatto per il problema proposto?