Università
degli Studi
di Ferrara

TUTORATO 02

PROGRAMMAZIONE E LABORATORIO A.A. 2021/2022

Dipartimento di Matematica ed Informatica

Elena M Galdi

CALENDARIO PROVVISORIO

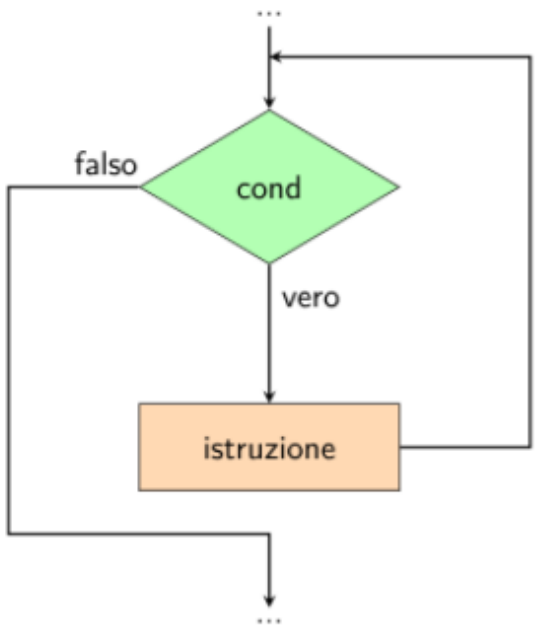
- Lunedì 11/10 : 8,30 – 10,30
- **Lunedì 18/10 : 8,30 – 10,30**
- Lunedì 25/10 : 8,30 – 10,30
- Venerdì 05/11 : 8,30 – 10,30
- Lunedì 08/11 : 8,30 – 10,30
- Lunedì 15/11 : 8,30 – 10,30
- Lunedì 22/11 : 8,30 – 10,30
- Lunedì 29/11 : 8,30 – 10,30
- Lunedì 06/12 : 8,30 – 10,30
- Lunedì 13/12 : 8,30 – 10,30

COMPILARE ED ESEGUIRE UN FILE .C DA TERMINALE

- COMPILAZIONE:
  - **Compila e crea file eseguibile a.out:** gcc *nome_file.c*
  - **Compila e crea file eseguibile nome_file:** gcc –o *nome_file nome_file.c*
  - **Compila per debug e crea file eseguibile a.out:** gcc –g *nome_file.c*
  - **Compila e crea file eseguibile nome_file:** gcc –g –o *nome_file nome_file.c*
- ESECUZIONE
  - *./a.out*
  - *./nome_file*

CONTROLLO DI FLUSSO: CICLO WHILE

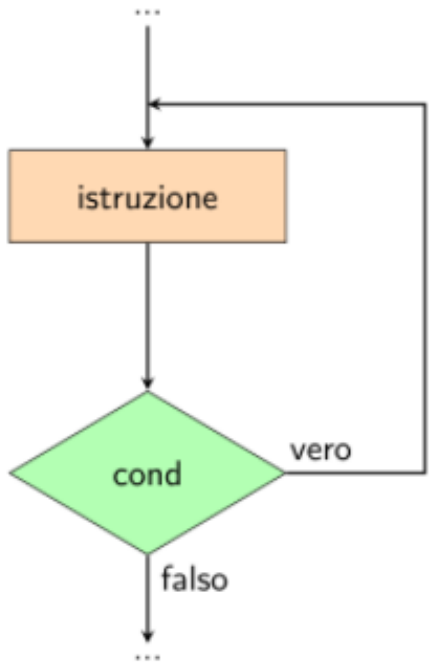
Diagramma di flusso `while`




```
while (__cond__)
    __istruzione__
```

CONTROLLO DI FLUSSO: CICLO DO - WHILE

Diagramma di flusso `do...while`

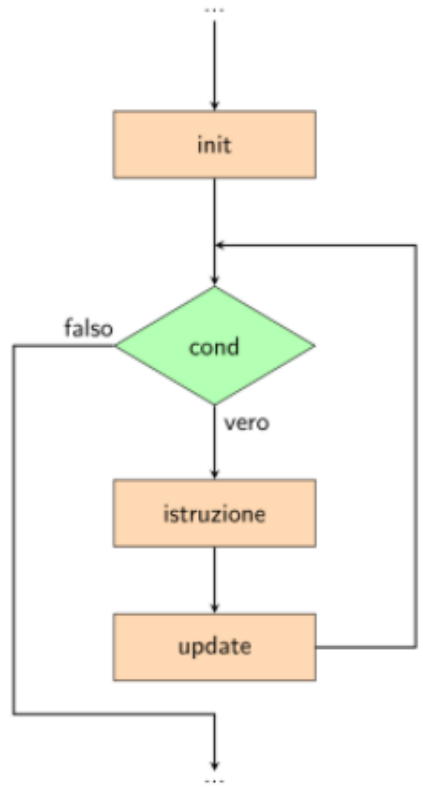



```
do __istruzione__
while (__cond__);
```

CONTROLLO DI FLUSSO: WHILE VS DO-WHILE

CONTROLLO DI FLUSSO: CICLO FOR

Diagramma di flusso `for`




```
for (__init__, __cond__, __update__)
```

```
__istruzione__
```

CONTROLLO DI FLUSSO: WHILE VS FOR

- Ciclo for: quando si conosce a priori il numero di iterazioni da fare

ESERCIZIO 1 : INPUT

- Leggere da tastiera una serie di numeri compresi tra 1 e 100 finché non si inserisce 0.
- Stampare poi a video:
  - La somma dei numeri inseriti.
  - Il numero più grande.
  - Il numero più piccolo.

ESERCIZIO 2: COLLATZ

- Sviluppare il seguente algoritmo:
  - Si prenda numero intero positivo n.
  - Se n = 1 l'algoritmo termina.
  - Finché *n* > 1, se n è pari, lo si divida per 2; se n dispari lo si moltiplichi per 3 e si aggiunga 1.

$$f(n) = \begin{cases} n/2 & \text{se n è pari} \\ 3n+1 & \text{se n è dispari} \end{cases}$$

- La congettura di Collatz asserisce che questo algoritmo giunge sempre a termine, indipendentemente dal valore di partenza.
- Leggere da tastiera due numeri interi *a* e *b* e, per tutti i numeri compresi tra *a* e *b*, stampare quanti cicli compie l'algoritmo

ESERCIZIO 3: NUMERO PERFETTO

- Sviluppare un algoritmo per trovare i numeri perfetti tra 1 e 10000. Un numero si dice perfetto se è uguale alla somma dei suoi divisori.
- Esempio: 28 = 1 + 2 + 4 + 7 + 14 è un numero perfetto

ESERCIZIO 4: ALGORITMO EUCLIDEO

- L'algoritmo di Euclide è utilizzato per calcolare il massimo comune divisore (mcd) tra due numeri interi. L'algoritmo è il seguente:
  - Siano a e b due interi con $0 \leq b < a$.
  - Se $b = 0$ allora $mcd(a, b) = a$.
  - Se $b \neq 0$ allora $a = b * q + r$ con $0 \leq r < b$, con q quoziente e r resto della divisione tra a e b.
  - Porre $a = b$ e $b = r$.
  - Ripartire dal punto 1.
- L'algoritmo continua finché non si trova un $b = 0$.
- Se $b = 0$ allora $mcd(a, b) = a$.
- Sviluppare un algoritmo che, presi in ingresso due interi a e b, ne calcoli il mcd.
  Es: $mcd(126, 147) = 21$

ESERCIZIO 5: PALINDROMO

$N == N_{rov}$

- Leggere da tastiera un numero intero (< 2147483647) e controllare se è palindromo.
- Un numero è palindromo quando le sue cifre rappresentano lo stesso valore sia che siano lette da destra che da sinistra.
  Es: 1234321 è palindromo

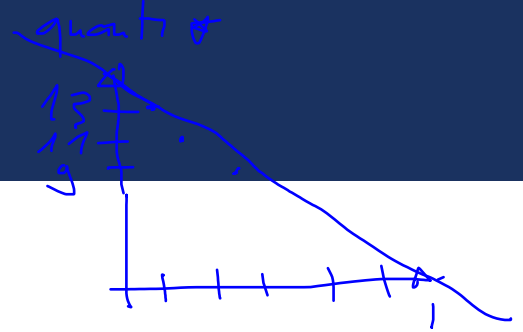
N
50105

50105

1234	%10	4	=4
123	%10=3	4·10+3	=43
12	%10=2	43·10+2	=432
1	%10=1	432·10+1	=4321

acc
N_rov

ESERCIZIO 6: PIRAMIDE



N = 7

- Leggere un intero compreso fra 1 e 40 e stampare una piramide di asterischi invertita di altezza pari al numero letto
  Ad esempio, se si inserisce 7, stampare quanto segue:

```
Inserire numero di righe: 7
* * * * * * * * * * * * * *
  * * * * * * * * * * * * *
    * * * * * * * * * * *
      * * * * * * * * *
        * * * * * * *
          * * * * *
            * * *
              *
```