

Costrutti di ripetizione in C

while - for - do/while

Titolo unità didattica: Costrutti di controllo

[04]

Titolo modulo : Costrutti di ripetizione in C

[04-C]

Analisi dei costrutti di ripetizione in C

Argomenti trattati:

- ✓ costrutto **while**
- ✓ costrutto **for**
- ✓ costrutto **do-while**
- ✓ costrutti di ripetizione nidificati

Prerequisiti richiesti: P1-03-03-C, P1-04-02-T

while

costrutti di ripetizione in C

WHILE

```
while (<predicato>)  
{  
    <blocco del ciclo>  
}
```

blocco

Finchè la condizione è vera esegue il blocco di codice.

Il blocco di codice viene ripetuto finchè la condizione risulta vera.

Esempio: finchè ho prodotti nella lista della spesa, compra il prossimo articolo e cancellalo dalla lista.

costrutti di ripetizione in C

WHILE

```
while (<predicato>)  
{  
    <blocco del ciclo>  
}
```

blocco

Quando la condizione diventa falsa si esce dal ciclo.

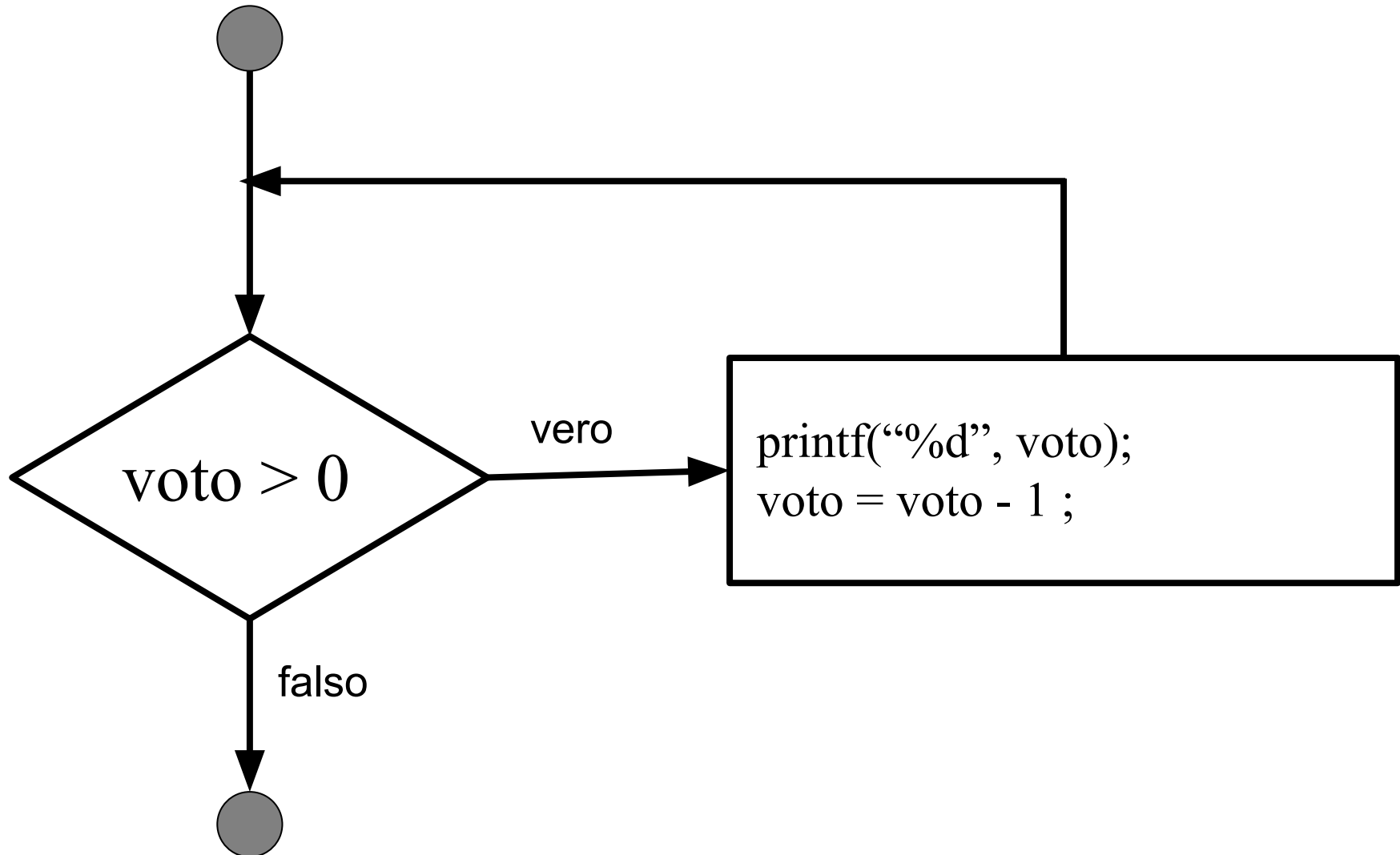
Esempio while.

Finchè il mio numero è maggiore di 0, sottrai 1

```
int voto;  
voto = 10;  
while (voto>0){  
    printf("%d \n", voto);  
    voto = voto -1 ;  
}  
  
return 0;
```

costrutti di ripetizione in C

WHILE



Il contatore.

L'iterazione controllata dal contatore utilizza una variabile detta contatore, per specificare il numero di volte che dovrà avvenire un'insieme di istruzioni dovrà essere eseguita.

Ad esempio: Stampa 10 volte *

```
int contatore = 10;
while (contatore > 0){
    printf("*");
    contatore = contatore -1;
}
```

Flag, o valore sentinella

Il flag dà all'utente la possibilità di decidere quando fermare l'iterazione.

Il valore sentinella dovrà essere scelto in modo che non sarà confuso con un valore di input accettabile. Di solito si usa -1

Ad esempio: somma tutti i valori positivi inseriti.

```
int voto = 0;
int totale = 0;

printf("Inserisci i voti da sommare \n");
scanf("%d", &voto);

while(voto != -1){
    totale = voto + totale;
    scanf("%d", &voto);
}
```

Operatori di assegnamento

Abbreviazioni per le espressioni

Operatore	Esempio di utilizzo	Spiegazione
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>
<code>-=</code>	<code>c -= 7</code>	<code>c = c - 7</code>
<code>*=</code>	<code>c *= 7</code>	<code>c = c * 7</code>
<code>/=</code>	<code>c /= 7</code>	<code>c = c / 7</code>
<code>%=</code>	<code>c %= 7</code>	<code>c = c % 7</code>
<code>++</code>	<code>++a</code>	incrementa a di 1 e quindi utilizza il nuovo valore di a
<code>++</code>	<code>a++</code>	utilizza il valore corrente di a e poi lo incrementa di 1
<code>--</code>	<code>--a</code>	decrementa a di 1 e quindi utilizza il nuovo valore di a
<code>--</code>	<code>a--</code>	utilizza il valore corrente di a e poi lo decrementa di 1
<code>?:</code>	<code>int voto = 0;</code> <code>(voto == 0)? printf("0") : printf("1");</code>	abbreviazione di if/else (condizione)? (istruzione se è vera) : (istruzione se è falsa)

for

—

costrutti di ripetizione in C

FOR

```
for (i=a; i<=b; i=i+c)
{
    <blocco del ciclo>
}
```

blocco

for gestisce automaticamente tutti i dettagli di una iterazione controllata da un contatore.

Esempio visto in precedenza (for VS while)

Ad esempio: Stampa 10 volte *

```
int contatore;  
  
for(contatore = 10; contatore > 0; contatore--){  
    printf("*");  
}
```

prima con while...

```
int contatore = 10;  
while (contatore > 0){  
    printf("*");  
    contatore = contatore -1;  
}
```

Esempio precedente, con incremento positivo

Ad esempio: Stampa 10 volte *

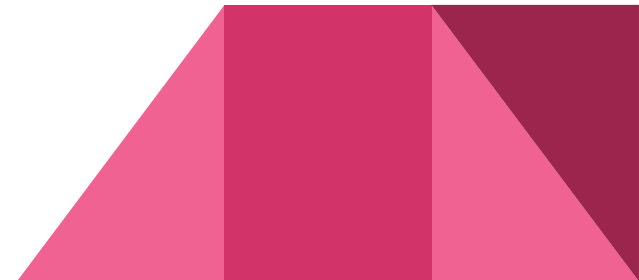
```
int contatore;  
  
for(contatore = 0 ; contatore < 10; contatore++){  
    printf("*");  
}
```

prima con while...

```
int contatore = 0;  
  
while (contatore < 10){  
    printf("*");  
    contatore++;  
}
```

Esempi intestazione con for

- a) Far variare la variabile di controllo da 1 a 100 con incrementi di 1
`for(i = 1; i <= 100; i++)`
- b) Far variare la variabile di controllo da 100 a 1 con decrementi di 1
`for(i = 100; i >= 1 ; i--)`
- c) Far variare la variabile di controllo da 7 a 77 a passi di 7
`for(i=7; i<= 77; i += 7)`
- d) Fara variare la variabile di controllo sulla seguente sequenza di valori 99, 88, 77, 66, 55, 44, 33, 22, 11, 0
`for(j= 99; j >= 0; j -= 11)`



il costrutto **for** in C è **equivalente** al **while**

```
for (i=a;i<=b;i=i+c)
{
    <blocco del ciclo>
}
```

sono equivalenti

```
i=a;
while (i <= b)
{
    <blocco del ciclo>
    i=i+c;
}
```

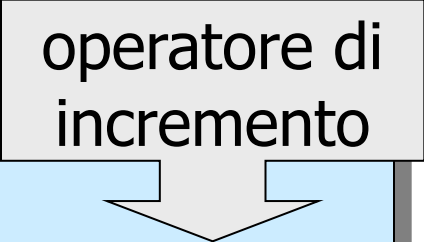
Esempio:

visualizzazione dei numeri interi da 1 a 10

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<=10 ; i=i+1)
    {
        printf("%3d",i) ;
    }
}
```

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
    {
        printf("%3d",i) ;
    }
}
```

operatore di incremento



1 2 3 4 5 6 7 8 9 10 _

Esempio:

visualizzazione dei numeri interi da 1 a 10

i blocchi possono essere omessi
nel caso in cui **<corpo>** sia
una unica istruzione

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
        printf("%3d",i) ;
}
```

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
    {
        printf("%3d",i) ;
    }
}
```

1 2 3 4 5 6 7 8 9 10_

Esempio:

visualizzazione dei numeri pari da -8 a 10

```
#include <stdio.h>
/* visualizza i numeri pari da -8 a 10 */
void main()
{
    int i;
    for(i=-8; i<=10 ; i=i+2)
        printf("%3d",i) ;
}
```

-8 -6 -4 -2 0 2 4 6 8 10_

do/while

costrutti di ripetizione in C

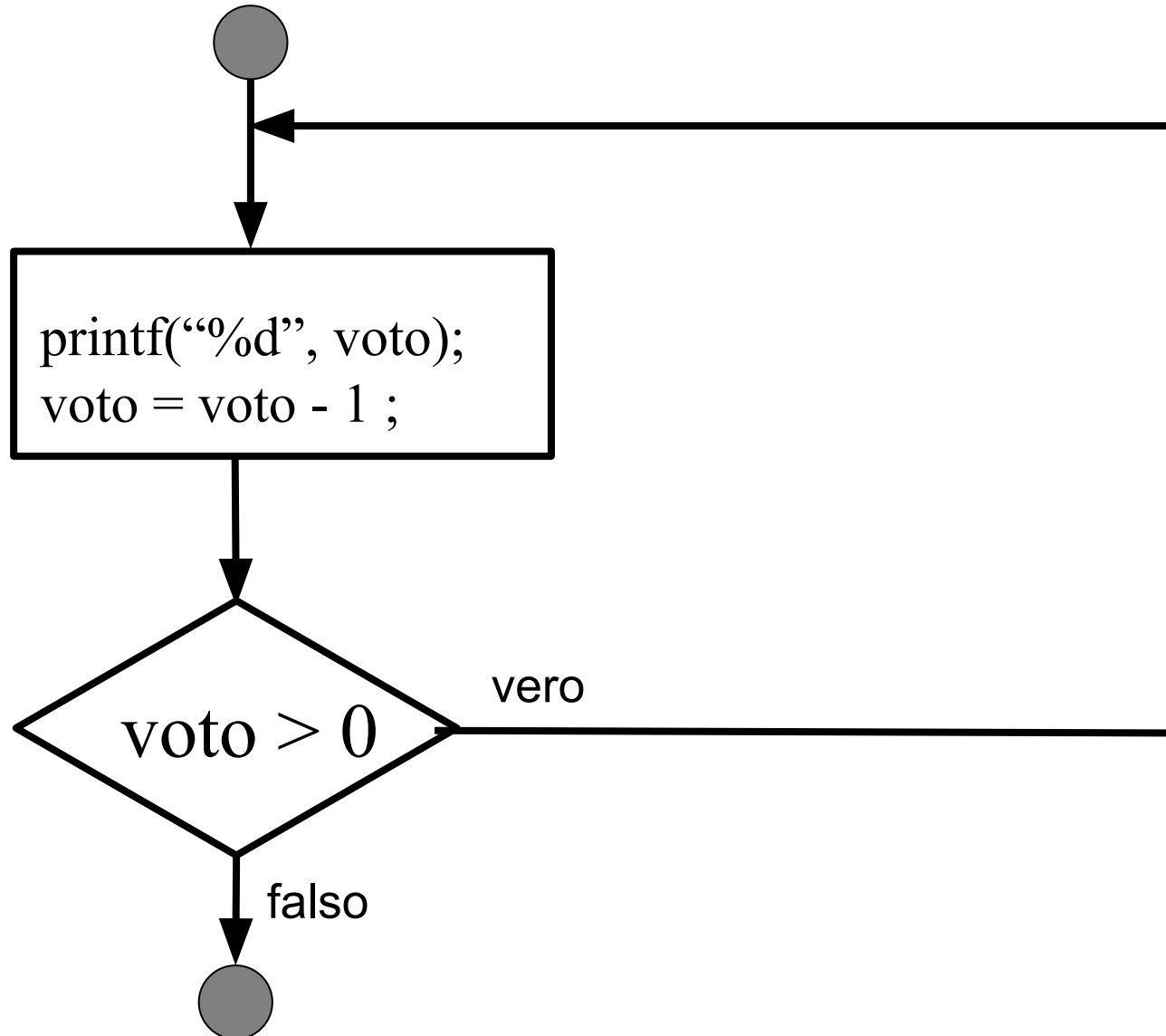
```
do
{
    <blocco del ciclo>
}
while (<predicato>);
```

blocco

Viene controllata la condizione dopo l'esecuzione del corpo.
Le istruzioni all'interno del corpo saranno eseguite almeno una volta.

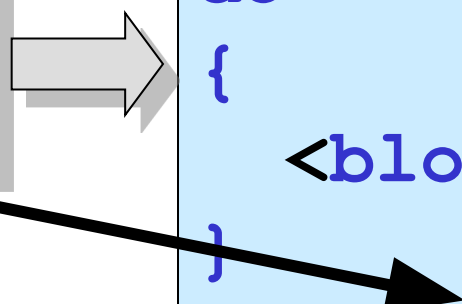
costrutti di ripetizione in C

DO/WHILE



costrutti di ripetizione in C

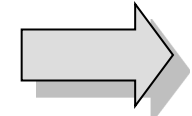
ripetere:
 <blocco del ciclo>
finché <predicato>



```
do
{
    <blocco del ciclo>
}
while( <predicato> );
```

Esempio

ripetere
 read (numero)
finché numero != 0



```
do
{
    scanf ("%d", &numero) ;
}
while (numero != 0) ;
```

Esempio:

visualizzazione dei numeri interi da 1 a 10

```
#include <stdio.h>
void main()
{
    int i=1;
    while(i<=10) {
        printf("%3d",i) ;
        i++;
    }
}
```

```
#include <stdio.h>
void main()
{
    int i=1;
    do{
        printf("%3d",i) ;
        i++;
    } while(i<=10) ;
}
```

1 2 3 4 5 6 7 8 9 10_

operatori incremento e decremento in C

++ operatore di incremento

-- operatore di decremento

si applicano solo
alle variabili

`i = i+1;`

`++i;`

`i++;`

incremento

`i = i-1;`

`--i;`

`i--;`

decremento

Attenzione: `++i` `i++` `--i` `i--` sono espressioni
(hanno un valore):

```
int i=1;
printf("%d\n", i++);
printf("%d\n", i);
```

1
2_

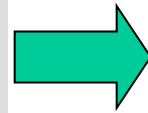
```
int i=1;
printf("%d\n", ++i);
printf("%d\n", i);
```

2
2_

Esempio:
visualizzazione dei primi 10 numeri pari

osservazione

primo numero pari: 2 , (2*1)
secondo numero pari: 4 , (2*2)
terzo numero pari: 6 , (2*3)
...



i -simo numero pari $2*i$

```
#include <stdio.h>
/* visualizza i primi 10 numeri pari */
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
        printf("%2d-simo numero pari:%d\n",i,2*i) ;
}
```

Esempio: visualizzazione dei primi 10 numeri pari

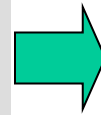
```
#include <stdio.h>
/* visualizza i primi 10 numeri pari */
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
        printf("%2d-simo numero pari:%d\n",i,2*i) ;
}
```

```
1-simo numero pari :2
2-simo numero pari :4
3-simo numero pari :6
4-simo numero pari :8
5-simo numero pari :10
6-simo numero pari :12
7-simo numero pari :14
8-simo numero pari :16
9-simo numero pari :18
10-simo numero pari :20
```

Esempio:
visualizzazione dei primi 10 numeri dispari

osservazione

primo numero dispari: 1 , $(2*1-1)$
secondo numero dispari: 3 , $(2*2-1)$
terzo numero dispari: 5 , $(2*3-1)$
...



i -simo numero dispari
 $2*i-1$

```
#include <stdio.h>
/* visualizza i primi 10 numeri dispari */
void main()
{
    int i;
    for(i=1; i<=10 ; i++)
        printf("%2d-simo numero dispari:%d\n",i,2*i-1);
}
```

Esempio:

programma di controllo di correttezza della risposta (inserita da tastiera) a una domanda

```
#include <stdio.h>
void main()
{
    int risposta;
    const int risposta_corretta = 10;
    do {
        printf("qual e' il log in base 2 di 1024?\n");
        scanf("%d",&risposta);
        if (risposta == risposta_corretta)
            printf("risposta corretta\n");
        else
            printf("risposta sbagliata, ripetere\n");
    } while (risposta != risposta_corretta);
}
```

nidificazione di cicli (cicli innestati)

```
#include <stdio.h>
void main()
{
    int i,j;
    for (i=1; i<=2; i++)
        for (j=1; j<=3; j++)
            printf("( %d,%d)  ",i,j) ;
}
```

(1,1) (1,2) (1,3) (2,1) (2,2) (2,3)_

nidificazione di cicli (cicli innestati)

```
#include <stdio.h>
void main()
{
    int i,j;
    for (j=1; j<=3; j++)
        for(i=1; i<=2; i++)
            printf("( %d,%d)  ",i,j) ;
}
```

(1,1) (2,1) (1,2) (2,2) (1,3) (2,3)_

Esercizi

```
int main (void){
    int m, n, i, j;
    /* disegna un rettangolo di asterischi
       che abbia n righe e m colonne
       n e m sono valori dati in input */
    printf("Inserisci due valori\n");
    scanf("%d %d", &n, &m);

    for(i = 0; i<n; i++){
        for(j = 0; j<m; j++){
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

```
int main (void){  
    int m, n, i, j;  
    /* disegna un triangolo di asterischi  
       che abbia base e altezza n  
       n è un valore dato in input */  
    printf("Inserisci un valore\n");  
    scanf("%d", &n);  
  
    for(i = 0; i<=n; i++){  
        //printf("*");  
        for(j = 0; j<i; j++){  
            printf("*");  
        }  
        printf("\n");  
    }  
}
```

Esercizio 1 - while

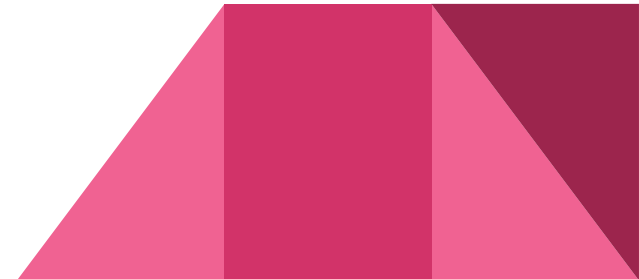
Una classe di 10 studenti sostiene un esame.

Avete a disposizione le votazioni per questo esame (numeri interi compresi tra 0 e 100).

Determinate la media della classe in questo esame.

Esempio per votazioni: 50, 90, 99, 18, 100, 45, 76, 81, 90, 61

Qual è la media della classe?



Esercizio 1 - while con flag

Una classe di studenti sostiene un esame, non so quanti studenti sono quindi usare un valore sentinella per terminare l'immissione di dati.

Avete a disposizione le votazioni per questo esame (numeri interi compresi tra 0 e 100).

Determinate la media della classe in questo esame.

Esempio per votazioni: 50, 90, 99, 18, 100, 45, 76, 81, 90, 61

Qual è la media della classe?



Esercizio 2

Riproduci il seguente codice utilizzando for al posto di while.

```
int voto = 0;
int totale = 0;

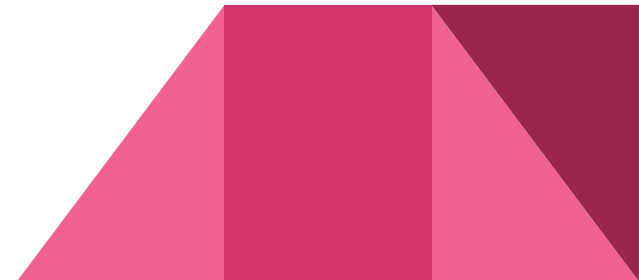
printf("Inserisci i voti da sommare \n");
scanf("%d", &voto);

while(voto != -1){
    totale = voto + totale;
    scanf("%d", &voto);
}
```

Esercizio 3

Scrivere un programma in C per visualizzare le seguenti sequenze di valori con for.

- a) 1, 2, 3, 4, 5, 6, 7
- b) 3, 8, 13, 18, 23
- c) 20, 14, 8, 2, -4, -10
- d) 19, 27, 35, 43, 51

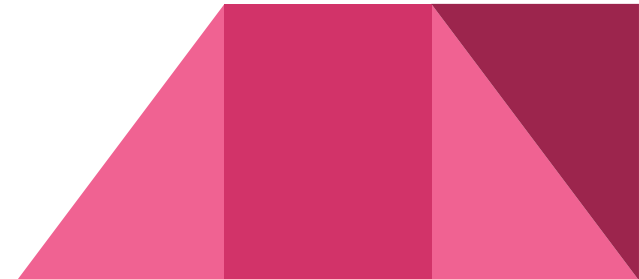


Esercizio 4 *

Scrivere un programma in C che richieda in input una serie di valori finchè non viene inserito dall'utente il valore -1.

Utilizzare uno dei costrutti di ripetizione studiati.

Alla fine del ciclo stampare il maggiore tra i numero inseriti.



Esercizio 5

Scrivere un programma che calcoli e visualizzi la somma degli interi pari da 2 a 30

Esercizio 6

Scrivere un programma che calcoli e visualizzi il prodotto degli interi dispari da 1 a 15

Esercizio 7

Scrivere un programma che trovi la media dei voti inseriti. Fare in modo che il primo valore inserito specifichi il numero di quelli ancora da leggere.

Non usare flag.

Esercizio 8

Scrivere un programma che visualizza **uno sotto l'altro** i seguenti disegni:

*	*****	*****	*
**	*****	*****	**
***	*****	*****	***
****	*****	*****	****
*****	*****	*****	*****
*****	*****	*****	*****
*****	*****	*****	*****
*****	****	****	*****
*****	***	***	*****
*****	**	**	*****
*****	*	*	*****

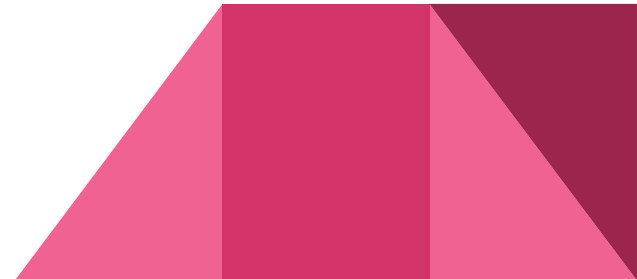




Esercizi di autovalutazione

Esercizio 1

Scrivere 4 differenti istruzioni in C che aggiungano 1 alla variabile intera



Esercizio 2

Scrivere una singola istruzione per ognuna delle seguenti attività:

- a) Assegnare la somma di x e di y a z e incrementare il valore di x di 1 dopo il calcolo.
- b) Moltiplicare la variabile product per 2 utilizzando *=
- c) Moltiplicare la variabile product per 2 utilizzando = e *
- d) Controllare se il valore della variabile count è maggiore di 10. Nel caso lo fosse visualizzare "Count maggiore di 10"
- e) Decrementare la variabile x di 1 e quindi sottrarla dalla variabile total
- f) Aggiungere la variabile x alla variabile total e quindi decrementare x di 1
- g) Calcolare il resto della divisione di q per divisor e assegnare il risultato a q. Scrivete questa istruzione in due modi diversi.
- h) Visualizzate il valore 123.4567 con 2 cifre di precisione. Quale valore sarà visualizzato?
- i) Visualizzate il valore in virgola mobile 3.14159 con tre cifre a destra del punto decimale. Quale valore sar visualizzato?

Esercizio 3

Scrivete una istruzione C per eseguire ognuna delle seguenti attività.

- a) dichiarare le variabili sum e x di tipo int
- b) inizializzate la variabile x con 1
- c) inizializzate la variabile sum con 0
- d) aggiungete la variabile x a sum e assegnate il risultato a quest'ultima
- e) visualizzate "la somma: " seguito dal valore della variabile sum

Esercizio 4

Combinare le istruzioni dell'esercizio precedente in un programma che calcoli la somma degli interi da 1 a 10. Utilizzate la struttura `while` per iterare le istruzioni di calcolo e di incremento. Il ciclo dovrà terminare quando il valore di `x` diventerà uguale a 1

Esercizio 5

Identificare e correggere gli errori in ognuna delle seguenti istruzioni:

- a)

```
while (c<=5){  
    product *= c;  
    c++;
```
- b)

```
scanf(".4%", &value);
```
- c)

```
if(gender ==1)  
    printf("Woman\n");  
else;  
    printf("Man\n");
```
- d)

```
while(z >= 0)  
    sum += z;
```



Soluzioni degli esercizi di autovalutazione

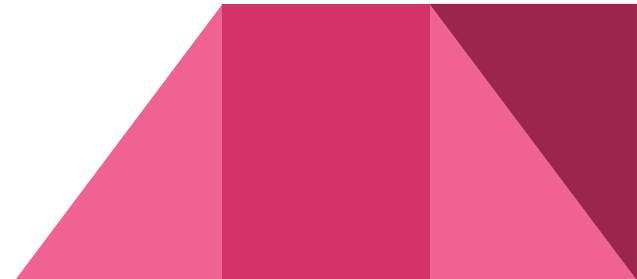
Esercizio 1

`x = x+1`

`x+=1`

`x++`

`++x`



Esercizio 2

- a) `z = x++ + y;`
- b) `product * = 2;`
- c) `product = product * 2;`
- d) `if(count < 0)`
 `printf("Count maggiore di 10\n");`
- e) `total -= -x;`
- f) `tola += x-;`
- g) `q %= divisor;`
 `q = q % divisor;`
- h) `printf("%.2f", 123.4567);`
 è visualizzato 123,46
- i) `printf("%.3f", 3.14159);`
 è visualizzato 3,142

Esercizio 3

- a) `int sum x;`
- b) `x = 1 ;`
- c) `sum = 0;`
- d) `sum += x; sum = sum +x;`
- e) `printf("La somma %d\n", sum);`

Esercizio 4

```
#include<stdio.h>
```

```
int main(){
```

```
    int sum, x;
```

```
    x=1;
```

```
    sum = 0;
```

```
    while (x <= 10){
```

```
        power *= x;
```

```
        ++1;
```

```
    }
```

```
    printf("%d", power);
```

```
    return 0;
```

```
}
```

Esercizio 5

- a) Errore: manca la parentesi graffa di chiusura del corpo del while
- b) Errore: è stata utilizzata la precisione nella specifica di conversione di scanf
- c) Errore: il punto e virgola dopo else produrrà un errore logico. La seconda printf sarà sempre eseguita
- d) Il valore della variabile z non sarà mai modificato all'interno della struttura while. E' stato quindi creato un ciclo infinito. Per evitare il ciclo infinito, z dovrà essere decrementata in modo da assumere, a un certo momento, il valore 0.