

Introduzione al C

Corso di Fondamenti di Informatica
Ingegneria Clinica

Esercitazione 3

DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

Domenico Daniele Bloisi
Raffaele Nicolussi

Esercizio

**Scrivere un programma che data in ingresso
una data in formato
giorno – mese – anno
stampi il giorno successivo**

Esempio

Se l'input è: 1 – 2 – 2007

il programma stampa: 2 – 2 – 2007

I commenti in c

Per commentare una singola riga inserire un doppio slash “//”

Per commentare più righe:

o si commentano le righe una per una

oppure si inserisce all’inizio della parte commentata “/*” e poi alla fine di essa “*/”

Esempio di commenti in c

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

int main()
{
    //chiedo all'utente di inserire un intero
    printf("Inserisci un numero:\n");
    int n;
    scanf("%d", &n);
    /* controllo che il numero immesso sia
    un numero positivo*/
    //in tal caso
    /*stampo una stringa di saluto*/
    if(n > 0)
        printf("bravo, sei un tipo positivo\n");
    system("PAUSE");
    return 1;
}
```

Le strutture di controllo

- If
- Operatore “?”
- Switch
- While
- Do-While
- For
- Break e Continue

Operatore if

condizione → if (expression)
statement ← **corpo: indentato
e racchiuso tra
graffe**

**L'if
ha 3 forme
di base**

```
if (expression)  
    statement1  
else  
    statement2
```

```
if (expression1)  
    statement1  
else if (expression2)  
    statement2  
else  
    statement3
```

Esempio if

```
main()
{
    int x, y, z;
    int w = 15;
    ...
    if (x < 0)
    {
        z=w;
        ...
    }
    else
    {
        y=w;
        ...
    }
}
```

condizione

corpo

Altro esempio if

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

int main()
{
    if(1)
        printf("dino\n");
    if(0)
        printf("gino\n");
    if(1256789)
        printf("nino\n");
    if(-874)
        printf("mino\n");
    system("PAUSE");
    return 1;
}
```

Quali nomi vengono stampati?

Attenzione a non mettere il “;”

```
if(0);  
{  
    printf("gino\n");  
}
```

Stampa
gino

Nell'if dopo le tonde se si inserisce il punto e virgola, l'istruzione che segue viene eseguita indipendentemente dall'esito del controllo

L'operatore “?”

L'operatore ? (ternary condition) e' la forma piu' **efficiente** per esprimere semplici if statements. Ha la seguente forma:

expression1 ? expression2 : expression3

che equivale a:

if expression1 then expression2 else expression3

Esempio “?”

`z = (a > b) ? a : b;`

cioe'

```
if (a > b)
```

```
    z = a;
```

```
else
```

```
    z = b;
```

assegna a z il massimo tra a e b.

**Un consiglio:
usatelo solo se
siete sicuri
di quello che state
facendo**

Switch

Permette scelte multiple tra un insieme di items. La sua forma generale e':

```
switch (expression) {  
    case item1:  
        statement1;  
        break;  
    .  
    .  
    case itemn:  
        statementn;  
        break;  
    case default:  
        statement;  
        break;  
}
```

Switch (2)

Il valore degli item deve essere una costante (**le variabili non sono permesse**).

Il break serve per terminare lo switch dopo l'esecuzione di una scelta, altrimenti verra' valutato anche il caso successivo.

E' possibile anche avere un'istruzione nulla, includendo solamente un ";"

oppure lasciando fallire l'istruzione di switch omettendo qualsiasi frase (vedi esempio di seguito).

Il caso "default" e' facoltativo e raggruppa tutti gli altri casi.

Esempio switch

```
switch (letter) {  
    case 'A':  
    case 'E':  
    case 'I':  
    case 'O':  
    case 'U':  
        num_vocali++;  
        break;  
    case " ":  
        num_spazi++;  
        break;  
    default:  
        num_altri++;  
        break;  
}
```

**num_vocali++;
equivale all'espressione
num_vocali = num_vocali + 1;**

**Si usa perché è più efficiente
e perché permette di minimizzare
eventuali errori di battitura**

**Il break termina la
valutazione**

Esempio switch (2)

Nel precedente esempio se `letter` e' una vocale ('A','E','I','O','U') viene incrementato il valore della variabile `num_vocali`, se e' uno spazio (" ") si incrementa `num_spazi`, altrimenti (se nessuno dei casi precedenti e' vero) viene eseguita la condizione di default e quindi viene incrementato `num_altri`.

Esercizio Switch

Chiedere all'utente di inserire un intero tra 1 e 12 (estremi inclusi, of course) e stampare il mese dell'anno ad esso corrispondente in formato non numerico

Esempio

Input: 2

Output: Febbraio

Soluzione Switch

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

int main()
{
    printf("inserisci un intero tra 1 e 12:\n");
    int x;
    scanf("%d", &x);
    switch(x)
    {
        case 1:
            printf("Gennaio\n");
            break;
        case 2:
            printf("Febbraio\n");
            break;
        case 12:
            printf("Dicembre\n");
            break;
        default:
            printf("c'hai provato...\n");
            printf("andra' meglio la prossima volta\n");
            break;
    }
    system("PAUSE");
    return 1;
}
```

Operatori incremento e decremento

In C è possibile utilizzare una forma abbreviata per indicare incrementi e decrementi

Incremento di 1

`variabile++` oppure `++variabile`

Quanto sopra equivale (entrambe le espressioni) a `variabile = variabile + 1`

Nel primo caso l'espressione in cui `variabile` si trova viene valutata prima che `variabile` sia incrementata, nel secondo caso prima viene incrementato il valore di `variabile` e poi viene valutata l'espressione in cui `variabile` si trova.

Operatori incremento e decremento (2)

```
int main(){
    int x = 1, y = 2;
    int z = y + x++;
    printf("z = %d x = %d\n", z, x);
    x = 1;
    z = y + ++x;
    printf("z = %d x = %d\n", z, x);
    return 1;
}
```

L'output è

```
z = 3   x = 2
z = 4   x = 2
```

Operatori incremento e decremento (3)

Vediamo ora il decremento

Decremento di 1

`variabile--` oppure `--variabile`

Quanto sopra (entrambe le espressioni) equivale a
`variabile = variabile - 1`

Come per l'incremento, nel primo caso l'espressione in cui variabile si trova viene valutata prima che variabile sia incrementata, nel secondo caso prima viene incrementato il valore di variabile e poi viene valutata l'espressione in cui variabile si trova.

Gli operatori +=, *=, /=, -=

Sempre al fine di minimizzare possibili errori di battitura, il c ha una forma compatta per le espressioni aritmetiche

+, -, *, /

L'espressione

variabile /= 3;

equivale a

variabile = variabile / 3;

L'istruzione while

L'istruzione "while" ha la seguente forma:

```
while (expression)
{
    statement;
}
```

 **Per le graffe vale la stessa regola vista per l'if**

Esempio while

```
main()
{
    int x = 3;
    while (x > 0)
    {
        printf("x = %d\n", x);
        x--;
    }
}
```

Questa istruzione è essenziale per la corretta esecuzione del programma. Se venisse omessa il programma non terminerebbe: ciclo infinito

Esempio while – Terminazione da utente

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

int main()
{
    printf("immetti un intero,\n");
    printf("sara' stampato il suo successore.\n");
    printf("123 per terminare\n");
    int x;
    scanf("%d", &x);
    while( x != 123)
    {
        printf("il successore e' %d\n\n\n", ++x);
        printf("immetti un intero,\n");
        printf("sara' stampato il suo successore.\n");
        printf("123 per terminare\n");
        scanf("%d", &x);
    }
    system("PAUSE");
    return 1;
}
```


Esercizio while

**Scrivere un programma che prenda in ingresso un intero positivo e stampi un numero di righe contenenti un asterisco ed il numero di riga nella forma seguente:
i. * dove i è il numero di riga.
(un numero non positivo per terminare)**

Esempio

Input: 2
Output: 1 . *
 2 . *

```

int main()
{
    printf("immetti un intero,\n");
    printf("sara' stampato un numero di righe.\n");
    printf("corrispondenti nella forma i. *\n");
    printf("dove i e' il numero di riga in ordine crescente\n");
    printf("(un numero non positivo per terminare)\n");
    int x;
    scanf("%d", &x);
    while( x > 0)
    {
        int counter = 0;
        while( x != 0)
        {
            printf("\n%d. *", ++counter);
            x--;
        }
        printf("\n\n\n");
        printf("immetti un intero,\n");
        printf("sara' stampato un numero di righe.\n");
        printf("corrispondenti nella forma i. *\n");
        printf("dove i e' il numero di riga in ordine crescente\n");
        printf("(un numero non positivo per terminare)\n");
        scanf("%d", &x);
    }
    system("PAUSE");
    return 1;
}

```

Soluzione while

il costrutto do-while

L'istruzione C "do-while" ha la seguente forma:

```
do  
    statement;  
while (expression);
```

**Un insieme di istruzioni
dopo il do: esso sarà
eseguito comunque
almeno una volta**

**notare i due punti
dopo le parentesi**

Esempio do-while

```
main()
{
    int x = 3;
    do {
        printf("x=%d\n",x);
        x--;
    }
    while (x > 0);
}
```

Il cui output e':

x=3

x=2

x=1

Istruzione for

il for è una istruzione equivalente dal punto di vista concettuale al while. Si tratta di quello che in gergo viene definito “zucchero sintattico”

```
for (expression1; expression2; expression3)
    statement;
    {or block of statements}
```

dove expression1 inizializza, expression2 e' il test di termine ed expression3 e' il modificatore (che puo' anche essere piu' di un semplice incremento).

Esempio for

```
main()
{
    int i;
    for ( i = 0; i < 3; i++)
        printf("i = %d\n", i);
}
```

che genera come output sullo schermo:

x = 0

x = 1

x = 2

Esercizio for

Calcolare con il costrutto for la somma di tutti i numeri da 0 ad n. Con n inserito dall'utente.

Si può risolvere con una semplice espressione matematica, quale?

Noi, giusto per esercizio, lo facciamo con il for e ringraziamo Gauss in ogni modo.

Soluzione for (stupida)

```
int main()
{
    printf("stupido esempio di uso del for\n");
    printf("Please, enter a positive non zero number:\n");
    int n;
    scanf("%d", &n);
    while(n <= 0)
    {
        printf("Please, enter a positive non zero number:\n");
        scanf("%d", &n);
    }
    printf("Thank you.\n");
    int sum = 0;
    int i;
    for(i = n; i > 0; i--)
        sum += i;
    printf("the sum of all the integers between 0 and %d, is %d\n", n, sum);

    system("PAUSE");
    return 1;
}
```


Soluzione (non for) intelligente

La somma S dei numeri interi da 0 ad n è

$$S = \frac{n}{2} (n + 1)$$

Il comando break

Lo abbiamo incontrato nel costrutto switch

il comando break serve per controllare i loop.

In particolare: esce da un loop o da uno switch

Il comando continue

il comando break serve per controllare i loop.

In particolare: salta una iterazione del loop.

Esempio break e continue

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

int main()
{
    int i;
    for(i = 0; i < 10; i++)
    {
        if(i == 3)
            continue;
        if(i == 7)
            break;
        printf("%d\n", i);
    }
    system("PAUSE");
    return 1;
}
```

Stampa:

1
2
4
5
6

Esercizio break e continue

Si chiede di leggere un valore integer ed elaborarlo in accordo con le seguenti condizioni:

Se il valore che abbiamo letto e' negativo, si vuole stampare un messaggio di errore ed abbandonare il loop.

Se il valore letto e' maggiore di 100, si vuole ignorarlo e continuare con il successivo valore in input.

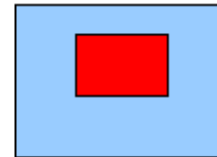
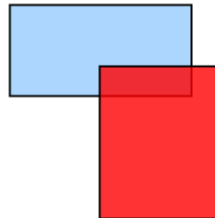
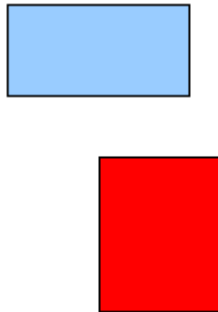
Se il valore e' 0, dovremo terminare il loop.

Soluzione break e continue

```
/* Viene letto un valore intero ed elaborato purchè
sia maggiore di 0 e minore di 100 */
while (scanf("%d", &value) == 1 && value !=0) {
    if (value < 0) {
        printf("Valore non ammesso\n");
        break; /* Abbandona il loop */
    }
    if (value > 100)
    {
        printf("Valore non ammesso\n");
        continue; /*Torna nuovamente all'inizio del loop */
    }
    /*Elabora il valore letto*/
    /*che e' sicuramente tra 0 e 100 */
    .
    .
    .
}
```

Esercizio

- Leggere le coordinate di due rettangoli R1 e R2 e stampare
 - 0 se i rettangoli sono disgiunti
 - 1 se si sovrappongono parzialmente
 - 2 se uno contiene l'altro



Soluzione parziale – solo ultimo caso

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

int main()
{
    printf("R1\n");
    printf("inserisci i valori (x, y) per il primo vertice\n"
           " separati da uno spazio:\n");
    int R1_x1, R1_y1;
    scanf("%d %d", &R1_x1, &R1_y1);
    printf("inserisci i valori (x, y) per il secondo vertice\n"
           " separati da uno spazio:\n");
    int R1_x2, R1_y2;
    scanf("%d %d", &R1_x2, &R1_y2);

    int R1_left_bottom_vertex_x, R1_left_bottom_vertex_y,
        R1_right_upper_vertex_x, R1_right_upper_vertex_y;

    if(R1_x1 > R1_x2)
    {
        R1_right_upper_vertex_x = R1_x1;
        R1_left_bottom_vertex_x = R1_x2;
    }
    else
    {
        R1_right_upper_vertex_x = R1_x2;
        R1_left_bottom_vertex_x = R1_x1;
    }
}
```


Soluzione parziale (continua)

```
if(R1_y1 > R1_y2)
{
    R1_right_upper_vertex_y = R1_y1;
    R1_left_bottom_vertex_y = R1_y2;
}
else
{
    R1_right_upper_vertex_y = R1_y2;
    R1_left_bottom_vertex_y = R1_y1;
}

printf("R2\n");
printf("inserisci i valori (x, y) per il primo vertice\n"
      " separati da uno spazio:\n");
int R2_x1, R2_y1;
scanf("%d %d", &R2_x1, &R2_y1);
printf("inserisci i valori (x, y) per il secondo vertice\n"
      " separati da uno spazio:\n");
int R2_x2, R2_y2;
scanf("%d %d", &R2_x2, &R2_y2);

int R2_left_bottom_vertex_x, R2_left_bottom_vertex_y,
    R2_right_upper_vertex_x, R2_right_upper_vertex_y;
```

Soluzione parziale (continua)

```
if(R2_x1 > R2_x2)
{
    R2_right_upper_vertex_x = R2_x1;
    R2_left_bottom_vertex_x = R2_x2;
}
else
{
    R2_right_upper_vertex_x = R2_x2;
    R2_left_bottom_vertex_x = R2_x1;
}

if(R2_y1 > R2_y2)
{
    R2_right_upper_vertex_y = R2_y1;
    R2_left_bottom_vertex_y = R2_y2;
}
else
{
    R2_right_upper_vertex_y = R2_y2;
    R2_left_bottom_vertex_y = R2_y1;
}
```

Soluzione parziale (finisce)

```
if((R2_right_upper_vertex_x < R1_right_upper_vertex_x) &&
   (R2_left_bottom_vertex_x > R1_left_bottom_vertex_x) &&
   (R2_right_upper_vertex_y < R1_right_upper_vertex_y) &&
   (R2_left_bottom_vertex_y > R1_left_bottom_vertex_y))
{
    printf("2 - R2 e' strettamente contenuto in R1\n");
}
else
    printf("R2 NON e' strettamente contenuto in R1\n");

system("PAUSE");
return 1;
}
```



**La stessa cosa
può esser fatta
con R1 ed R2
invertiti**