

Esercizi di programmazione in linguaggio C - Costrutto selezione

prof. Roberto Fuligni

1. **[confronto1]** - Scrivere un programma che, richiesti in ingresso due valori interi distinti, ne determini il maggiore.
2. **[confronto2]** ★ Scrivere un programma che, richiesti due numeri interi qualsiasi, segnali se i due numeri sono uguali; in caso contrario indichi il minore e il maggiore.
3. **[sconto]** - Un negoziante per ogni spesa di importo superiore a 100 € effettua uno sconto del 5%, del 10% per ogni spesa superiore a 300 €. Scrivere un programma che richieda all'utente l'ammontare della spesa e visualizzi quindi l'importo effettivo da pagare.
4. **[trenumeri]** - Scrivere un programma che, richiesti in input tre numeri interi, visualizzi a seconda dei casi una delle seguenti risposte:
 - Tutti uguali
 - Due uguali e uno diverso
 - Tutti diversi
5. **[tipotriangolo]** - Scrivere un programma che, lette le misure dei lati di un triangolo, visualizzi se il triangolo è equilatero, isoscele o scaleno.
6. **[biglietto]** ★ Su una linea ferroviaria, rispetto alla tariffa piena, gli utenti pensionati usufruiscono di uno sconto del 10%, gli studenti del 15% e i disoccupati del 25%. Codificando i pensionati con una P, gli studenti con una S e i disoccupati con una D, scrivere un programma che, richiesto il costo di un biglietto e l'eventuale condizione particolare dell'utente, visualizzi l'importo da pagare.
7. **[menuscelta]** ★ Scrivere un programma che visualizzi il seguente menu:

MENU DI PROVA - premere:

- a) Per immettere dati
- b) Per determinare il maggiore
- c) Per determinare il minore
- d) Per ordinare
- e) Per visualizzare

Scelta: _

attenda l'immissione di un carattere da parte dell'utente e infine, usando il costrutto *switch*, visualizzi una scritta corrispondente alla scelta effettuata (per esempio del tipo: In esecuzione l'opzione a).

Se la scelta non è tra quelle proposte (a,b,c,d,e) deve invece essere visualizzata la scritta *Opzione inesistente*.

8. **[equazione1g]** ★ Scrivere un programma che risolva l'equazione di primo grado $ax = b$, visualizzando a seconda dei casi la soluzione, la scritta *equazione indeterminata* oppure *equazione impossibile*. I coefficienti reali *a* e *b* devono essere richiesti in input all'utente.

9. **[equazione2g]** - Scrivere un programma per risolvere l'equazione di secondo grado $ax^2 + bx + c = 0$, essendo a, b, c coefficienti reali richiesti in input all'utente. Il programma deve comunicare, a seconda dei casi, le due soluzioni oppure la scritta `Non esistono soluzioni reali`.
10. **[idraulico]** - Un idraulico chiede € 40,00 per un'ora di lavoro, più il costo del materiale, con un minimo di € 100,00 per ogni lavoro. Scrivere un programma che, dati in input il costo del materiale e il numero di ore lavorative, determini la spesa totale, facendola ammontare al limite minimo quando previsto.
11. **[misura]** ★ Durante un esperimento di fisica, si ripete la misura del tempo di oscillazione di un pendolo per tre volte, ottenendo valori di volta in volta diversi. Per questo motivo, si decide di esprimere la misura attraverso il valor medio e di indicare l'incertezza mediante la semidispersione massima, cioè la semidifferenza tra i valori massimo e minimo: $(t_{\max} - t_{\min}) / 2$.
Scrivere un programma che, richiesti in input i tre tempi in secondi, visualizzi la misura completa nel formato:

$$t_{\text{medio}} \pm \text{incertezza}$$

Per esempio, inserendo i tempi: 4.5, 4.7, 4.6, l'output del programma deve essere: (4.6 ± 0.1) s.

12. **[giudizio]** – Nell'ambito di una data disciplina, uno studente effettua tre verifiche e ottiene, per ciascuna di esse, un voto espresso con un numero intero. Alla fine dell'anno, l'insegnante determina la media aritmetica dei tre voti (la media è in generale un numero reale) e formula un giudizio sulla base delle seguenti regole:

Valore della media	Giudizio espresso
$M < 4.5$	Gravemente insufficiente
$4.5 \leq M < 6.0$	Insufficiente
$6.0 \leq M < 7.5$	Sufficiente
$M \geq 7.5$	Buono

Scrivere un programma che richieda l'inserimento da tastiera dei tre voti e, dopo averne determinato la media, mostri il giudizio corrispondente.

13. **[gita]** – In una certa scuola le visite d'istruzione sono organizzate secondo il seguente schema:
- **Classi prime:** tutte le prime visiteranno Il museo egizio di Torino
 - **Classi seconde:** la 2^A visiterà il Duomo di Milano, tutte le altre seconde visiteranno il Castello Sforzesco di Milano
 - **Classi terze:** tutte le terze visiteranno le Gallerie degli Uffizi di Firenze
 - **Classi quarte:** la 4^B visiterà il Colosseo a Roma, la 4^C il Pantheon a Roma, tutte le altre quarte i Musei Vaticani.
 - **Classi quinte:** tutte le quinte visiteranno il Cern di Ginevra.

Scrivere un programma che richieda la classe (un numero intero tra 1 e 5), la sezione (una lettera maiuscola) e indichi la meta prevista sulla base dello schema precedente.

14. [**controllo data**] – Scrivere un programma che richieda l’inserimento da tastiera di tre numeri interi **g, m, a** (si ipotizza che **a** sia sempre un numero dispari). Successivamente il programma deve indicare se i tre numeri formano una data valida nel formato “g/m/a”.

Esempio: inserendo rispettivamente i valori 3, 5, 2017 oppure 30, 4, 2003, il messaggio da riportare è "DATA VALIDA" in quanto esistono i giorni 3/5/2017 e 30/4/2003.

Viceversa, inserendo per esempio i valori 33, 7, 2015, oppure 27, 14, 2011, il messaggio da visualizzare è "DATA NON VALIDA" perché i giorni 33/7/2015 e 27/14/2011 non esistono.

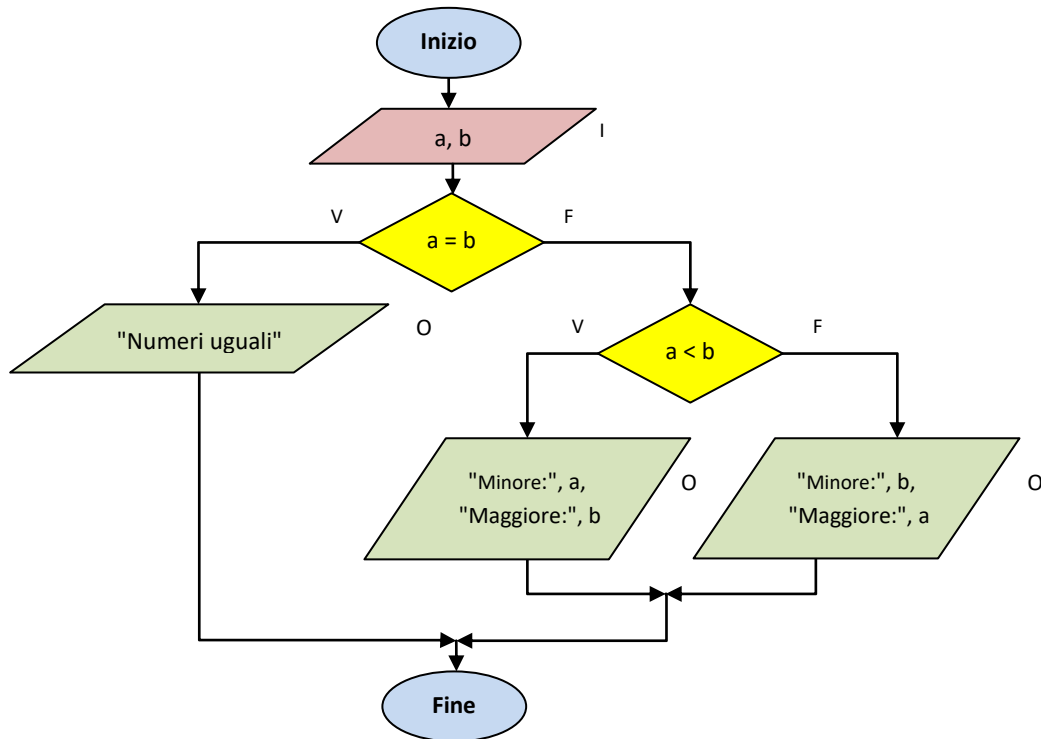
Nota: si ricorda che il dato **a**, in questo problema, è sempre dispari: gli anni dispari non sono mai bisestili.

15. [**triangolo rett**] ★ Scrivere un programma che, richieste in input le misure **interi** dei lati di un triangolo, indichi se il triangolo è rettangolo.
16. [**muro**] – Il Muro di Berlino è stato demolito nel 1989. Scrivere un programma che richieda l’anno di nascita dell’utente e dica se tale utente è nato nell’anno della caduta del Muro oppure quanti anni prima o dopo la caduta (per esempio, se l’utente è nato nel 1982, il messaggio da visualizzare è “L’utente è nato 7 anni prima della caduta del Muro di Berlino”).

Soluzioni di alcuni esercizi

Esercizio n. 2 (confronto2)

Diagramma a blocchi



Programma

```

/*
 * confronto2.c
 *
 * Segnala se due numeri interi dati in input sono uguali.
 * In caso contrario, indica il maggiore e il minore.
 */

#include <stdio.h>

int main(void) {
    int a, b;

    printf("Inserire due numeri interi: ");
    scanf("%d%d", &a, &b);

    if (a == b) {
        printf("I due numeri sono uguali\n");
    } else {
        if (a < b) {
            printf("Minore: %d Maggiore: %d\n", a, b);
        } else {
            printf("Minore: %d Maggiore: %d\n", b, a);
        }
    }
    return 0;
}

```

Esercizio n. 6 (biglietto)

Pseudocodifica

ALGORITMO biglietto

VARIABILI

```
costo_iniz: REALE
categoria: CARATTERE;
perc_sconto, sconto: REALE
importo: REALE
```

COSTANTI

```
PERC_SCONTO_PENSIONATI = 0.10
PERC_SCONTO_STUDENTI = 0.15
PERC_SCONTO_DISOCCUPATI = 0.25
```

INIZIO

```
SCRIVI("Costo iniziale del biglietto (Euro): ")
LEGGI(costo_iniz)
```

```
SCRIVI("P - Pensionato")
SCRIVI("S - Studente")
SCRIVI("D - Disoccupato")
SCRIVI("A - Altro")
SCRIVI("Categoria utente (P/S/D/A): ")
```

```
LEGGI(categoria)
```

```
NEL CASO CHE categoria SIA
```

```
  'P': perc_sconto ← PERC_SCONTO_PENSIONATI
  'S': perc_sconto ← PERC_SCONTO_STUDENTI
  'D': perc_sconto ← PERC_SCONTO_DISOCCUPATI
```

```
ALTRIMENTI
```

```
  perc_sconto ← 0
```

```
FINECASO
```

```
sconto ← perc_sconto * costo_iniz;
importo ← costo_iniz - sconto;
```

```
SCRIVI("Importo scontato (Euro): ", importo)
```

FINE

Programma

```
/*
 * biglietto.c
 *
 * Calcola il costo di un biglietto ferroviario
 * applicando uno sconto in base alla tipologia di utente
 */

#include <stdio.h>

#define PERC_SCONTO_PENSIONATI 0.10
#define PERC_SCONTO_STUDENTI 0.15
#define PERC_SCONTO_DISOCCUPATI 0.25

int main(void) {
    float costo_iniz;
    char categoria;
    float perc_sconto, sconto;
    float importo;

    printf("Costo iniziale del biglietto (Euro): ");
    scanf("%f", &costo_iniz);

    printf("\nP - Pensionato\nS - Studente\n");
    printf("D - Disoccupato\nA - Altro\n");
    printf("\nCategoria utente (P/S/D/A): ");

    // Si noti il carattere di newline, prima del placeholder %c,
    // per rimuovere dal buffer il "ritorno a capo"

    scanf("\n%c", &categoria);

    switch (categoria) {
        case 'P':
            perc_sconto = PERC_SCONTO_PENSIONATI;
            break;

        case 'S':
            perc_sconto = PERC_SCONTO_STUDENTI;
            break;

        case 'D':
            perc_sconto = PERC_SCONTO_DISOCCUPATI;
            break;

        default:
            perc_sconto = 0.0;
            break;
    }

    sconto = perc_sconto * costo_iniz;
    importo = costo_iniz - sconto;

    printf("\nImporto scontato (Euro): %8.2f\n", importo);

    return 0;
}
```

Esercizio n. 7 (menuscelta)

Programma

```
/*
 * menuscelta.c
 *
 * Mostra un menu' di scelte.
 * Esegue un'operazione in base alla scelta effettuata
 *
 */

#include <stdio.h>

int main(void) {

    char scelta;

    printf("MENU DI PROVA - premere:\n\n");
    printf("a) Per immettere dati\n");
    printf("b) Per determinare il maggiore\n");
    printf("c) Per determinare il minore\n");
    printf("d) Per ordinare\n");
    printf("e) Per visualizzare\n");
    printf("\n\tScelta: ");

    scanf("%c", &scelta);

    switch (scelta) {
        case 'a':
            printf("\nIn esecuzione l'opzione a\n");
            break;
        case 'b':
            printf("\nIn esecuzione l'opzione b\n");
            break;
        case 'c':
            printf("\nIn esecuzione l'opzione c\n");
            break;
        case 'd':
            printf("\nIn esecuzione l'opzione d\n");
            break;
        case 'e':
            printf("\nIn esecuzione l'opzione e\n");
            break;
        default:
            printf("\nOpzione inesistente\n");
            break;
    }

    return 0;
}
```

Esercizio n. 8 (equazione1g)

Analisi del problema

Il risultato prodotto in output dipende dal valore assunto dal coefficiente a :

- se $a \neq 0$, l'equazione ammette soluzione $x = \frac{b}{a}$
- se $a = 0 \wedge b = 0$, l'equazione è indeterminata
- se $a = 0 \wedge b \neq 0$, l'equazione è impossibile

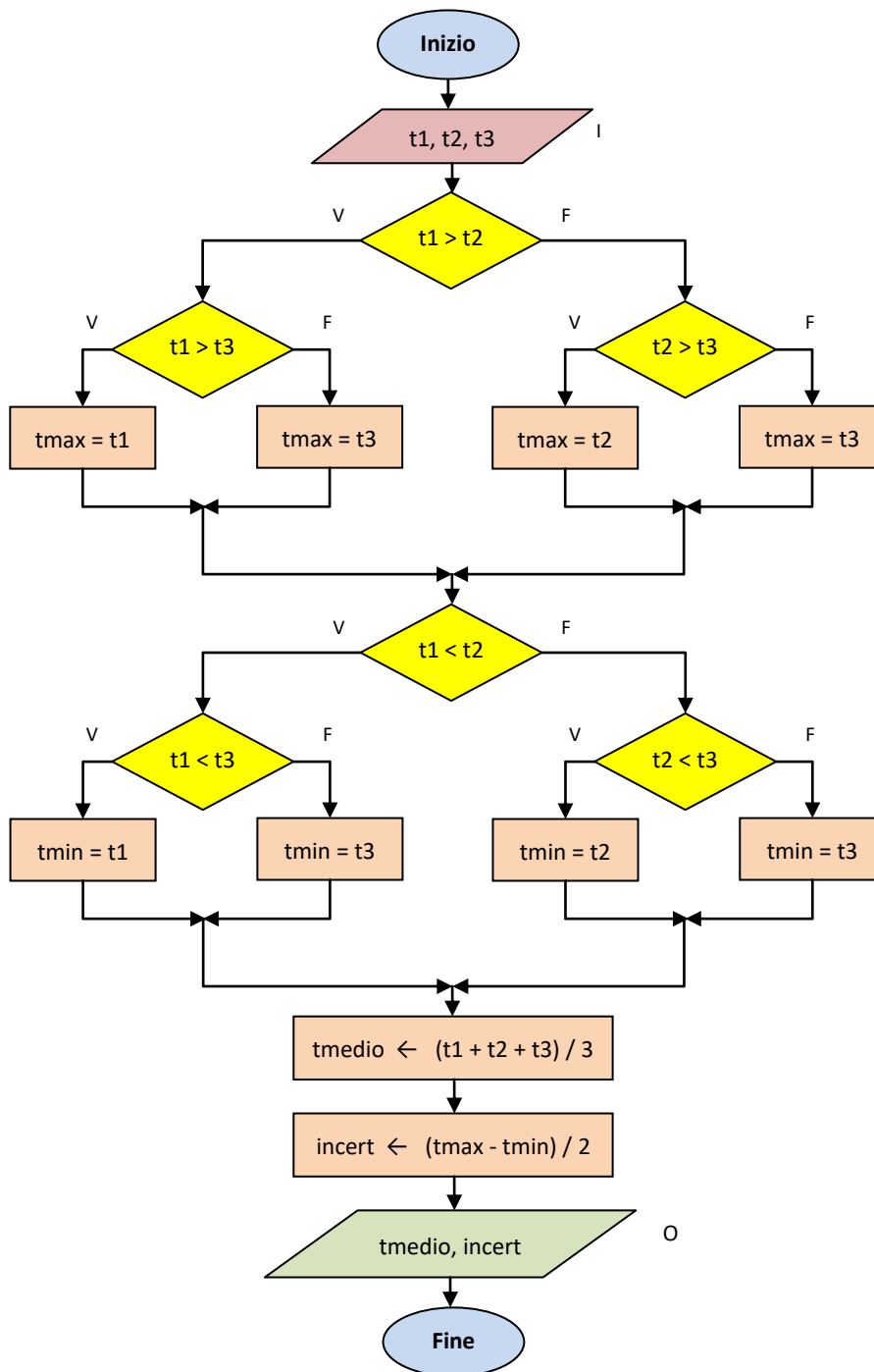
Specifiche funzionali

Dati iniziali:	Dati intermedi:	Dati finali:
a, b: coefficienti dell'equazione		x: soluzione dell'equazione

Pseudocodifica	Programma
<pre> ALGORITMO equazione1g VARIABILI a, b: REALE x: REALE INIZIO SCRIVI("Coefficienti a,b: ") LEGGI(a, b) SE a <> 0 ALLORA x ← b / a SCRIVI("Soluzione: ", x) ALTRIMENTI SE b = 0 ALLORA SCRIVI("Eq. indeterminata") ALTRIMENTI SCRIVI("Eq. impossibile") FINE </pre>	<pre> /* * equazione1g.c * * Determina la soluzione dell'equazione * di primo grado ax = b * */ #include <stdio.h> int main(void) { double a,b; double x; printf("Coefficienti a,b dell'equazione ax=b: "); scanf("%lf %lf", &a, &b); if (a != 0) { x = b / a; printf("Soluzione x = %lf\n", x); } else { if (b == 0) { printf("Equazione indeterminata"); } else { printf("Equazione impossibile"); } } return 0; } </pre>

Esercizio n. 11 (misura)

Diagramma a blocchi



Programma

```

/* misura.c
 *
 * Determina la migliore stima e l'incertezza di tre misure di tempo
 */

#include <stdio.h>

int main(void) {
    double t1, t2, t3;
    double tmin, tmax, tmedio, incert;

    printf("Inserire i tre valori di tempo (s): ");
    scanf("%lf%lf%lf", &t1, &t2, &t3);

    // Determinazione del valore massimo

    if (t1 > t2) {
        if (t1 > t3) {
            tmax = t1;
        } else {
            tmax = t3;
        }
    } else {
        if (t2 > t3) {
            tmax = t2;
        } else {
            tmax = t3;
        }
    }

    // Determinazione del valore minimo

    if (t1 < t2) {
        if (t1 < t3) {
            tmin = t1;
        } else {
            tmin = t3;
        }
    } else {
        if (t2 < t3) {
            tmin = t2;
        } else {
            tmin = t3;
        }
    }

    // Calcolo del valore medio e dell'incertezza assoluta

    tmedio = (t1 + t2 + t3) / 3.0;
    incert = (tmax - tmin) / 2.0;

    // Per visualizzare correttamente il simbolo ± nella finestra
    // di console, utilizzare la sequenza di escape \xF1 (il carattere
    // ha codice ASCII 241, corrispondente a F1 in esadecimale)

    printf("Misura: (%.2lf \xF1 %.2lf) s\n", tmedio, incert);
    return 0;
}

```

Esercizio n. 15 (triangolorett)

Analisi del problema

In un triangolo rettangolo di ipotenusa i e cateti c_1 e c_2 , l'ipotenusa è il lato di lunghezza maggiore. Per il teorema di Pitagora, vale inoltre la relazione:

$$i^2 = c_1^2 + c_2^2$$

da cui segue:

$$i^2 - c_1^2 - c_2^2 = 0$$

Un triangolo è rettangolo se, detti i il lato più lungo e c_1 e c_2 gli altri due lati, l'espressione $i^2 - c_1^2 - c_2^2$ è uguale a zero.

Specifiche funzionali

Dati iniziali:	Dati intermedi:	Dati finali:
a, b, c: lunghezze intere dei lati del triangolo	espr: risultato dell'espressione $i^2 - c_1^2 - c_2^2$	Nessuno

Pseudocodifica

```

ALGORITMO triangolorett
VARIABILI
    a, b, c: INTERO
    espr: INTERO
INIZIO
    SCRIVI("Inserire le lunghezze intere dei tre lati: ")
    LEGGI(a, b, c)

    SE (a > b) E (a > c) ALLORA
        espr ← a2 - b2 - c2
    ALTRIMENTI
        SE (b > a) E (b > c) ALLORA
            espr ← b2 - a2 - c2
        ALTRIMENTI
            espr ← c2 - a2 - b2

    SE espr = 0 ALLORA
        SCRIVI("Il triangolo è rettangolo")
    ALTRIMENTI
        SCRIVI("Il triangolo non è rettangolo")

FINE

```

Programma

```
/*
 * triangolorett.c
 *
 * Indica se un triangolo e' rettangolo
 * date le misure intere dei suoi lati.
 *
 */

#include <stdio.h>

int main(void) {
    int a, b, c;
    int espr;

    printf("Inserire le lunghezze intere dei tre lati: ");
    scanf("%d%d%d", &a, &b, &c);

    if (a > b && a > c)
        espr = a*a - b*b - c*c;          /* Il lato a e' il piu' lungo e quindi potrebbe */
                                        /* essere l'ipotenusa */
    else {
        if (b > a && b > c)
            espr = b*b - a*a - c*c;      /* Il lato b e' il piu' lungo */
        else
            espr = c*c - a*a - b*b;      /* Il lato c e' il piu' lungo */
    }

    if (espr == 0)
        printf("Il triangolo e' rettangolo\n");
    else
        printf("Il triangolo non e' rettangolo\n");

    return 0;
}
```