

ESERCIZIO 1

(Costrutti condizionali + Casting delle variabili)

Realizzare un programma che, dato in ingresso un angolo specificato in gradi come un numero intero, fornisca la relativa conversione in radianti. L'angolo deve essere compreso tra 0 e 360 gradi, altrimenti il programma stampa un messaggio di errore e termina.

Nota: Si consideri 3.14159265 il valore di pi greco.

```
#include <iostream>
#include <cstdlib>
using namespace std;

#define PIGRECO 3.14159265
// uso la direttiva #define per la costante PIGRECO
// alternativamente si poteva fare:
//                                     const double PIGRECO = 3.14159265
// Quali sono le differenze ?

int main(int argc, char *argv[])
{
    int gradi;
    float rad;
    cout << "\nInserire un angolo compreso tra 0 e 360 gradi: ";
    cin >> gradi;

    if (gradi < 0 || gradi > 360) // numero fuori range
    {
        cout << "\nErrore: inserito un angolo fuori range [0,360]";
    }
    else
    {
        rad = ((float) gradi) / 180.0 * PIGRECO;
        // NB.1: casting esplicito (poteva anche essere omesso)
        /* NB.2: per poter usare l'operatore '/' tra numero NON interi
           e' necessario che almeno uno dei due operandio sia una
           variabile di tipo float o double. */

        cout << "\n\nL'angolo inserito di " << gradi;
        cout << " gradi corrisponde a " << rad <<" radianti";
    }

    cout << "\n\nFINE ELABORAZIONE ----- \n";

    system("PAUSE");
    return EXIT_SUCCESS;
} // end main
```

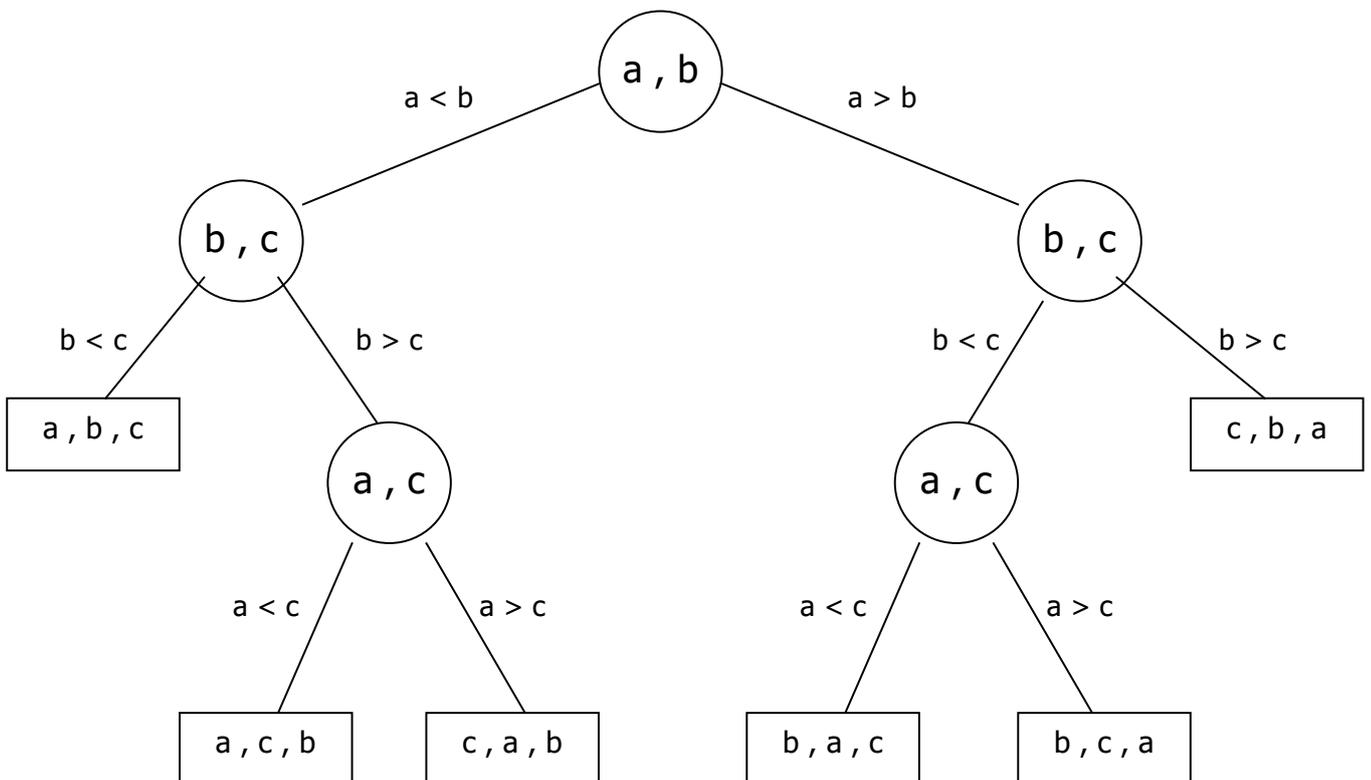
ESERCIZIO 2 (Costrutti condizionali + Algoritmi)

Scrivere un programma che risolva il seguente problema.
Letti tre numeri interi a , b , c dallo standard input, stampare a terminale la sequenza dei tre numeri in ordine non decrescente.
Esempio: $a = 10$, $b = 7$, $c = 9$ deve dare in uscita 7 9 10.

1° soluzione

Stesura informale dell'algoritmo:

Si può sicuramente condurre un'analisi per casi e, in base a quello individuato, scrivere le tre variabili in ordine non decrescente.



```
#include <iostream> // inclusione della libreria standard
#include <cstdlib> // per la funzione system(...)
using namespace std;
int main(int argc, char *argv[]) {
    int a,b,c; // dichiarazione delle variabili

    // legge tre interi a,b,c dallo standard input
    cout << "\n Inserisci il numero a: ";
    cin >> a;
    cout << "\n Inserisci il numero b: ";
    cin >> b;
    cout << "\n Inserisci il numero c: ";
    cin >> c;

    if (a < b) {

        if (b < c) {
            cout << "\n L'ordine voluto e': ";
            cout << a << " " << b << " " << c;
        }
        else {
            if (a < c) {
                cout << "\n L'ordine voluto e': ";
                cout << a << " " << c << " " << b;
            }
            else {
                cout << "\n L'ordine voluto e': ";
                cout << c << " " << a << " " << b;
            }
        }
    }
    else {
        if (b > c) {
            cout << "\n L'ordine voluto e': ";
            cout << c << " " << b << " " << a;
        }
        else {
            if (c > a) {
                cout << "\n L'ordine voluto e': ";
                cout << b << " " << a << " " << c;
            }
            else {
                cout << "\n L'ordine voluto e': ";
                cout << b << " " << c << " " << a;
            }
        }
    }
    cout << "\n\n FINE ELABORAZIONE....\n";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2° soluzione

Una seconda strategia di soluzione facilmente generalizzabile consiste nello scambiare ordinatamente le tre variabili finché i loro contenuti non risultino ordinati.

A tal fine quindi sarà necessaria sicuramente almeno un'altra variabile intera strumentale allo scambio tra due variabili.

Stesura informale dell'algoritmo:

1. Inizio del programma
2. Leggi i tre numeri a , b , c
3. confronta i valori di a e b , se non sono ordinati si effettua lo scambio
4. confronta i valori di a e c , se non sono ordinati si effettua lo scambio
5. confronta i valori di b e c , se non sono ordinati si effettua lo scambio
6. Stampa a video delle tre variabili a , b , c

Nota:

i punti **3**, **4**, **5** risolvono ciascuno lo stesso tipo di *sottoproblema* che è quello di ordinare i valori di due variabili.

La seconda soluzione pur essendo più semplice della soluzione 1, dal punto di vista della struttura e quindi anche della leggibilità, è meno efficiente: effettua sempre tre confronti diversamente dalla prima soluzione che ne effettua due in due casi su sei.

A ben vedere quindi la soluzione 1 risulta essere la soluzione *ottima*: non è possibile trovarne una che effettui un numero inferiore di confronti.

```
#include <iostream> // inclusione della libreria standard
#include <cstdlib> // per la funzione system(...)
using namespace std;
int main(int argc, char *argv[])
{
    int a,b,c,t; // dichiarazione delle variabili

    // legge tre interi a,b,c dallo standard input

    cout << "\n Inserisci il numero a: ";
    cin >> a;
    cout << "\n Inserisci il numero b: ";
    cin >> b;
    cout << "\n Inserisci il numero c: ";
    cin >> c;

    // ordinamento dei valori delle variabili a,b
    if (a > b)
    {
        // Scambio dei valori delle due variabili a,b
        t = a;
        a = b;
        b = t;
    }
    // ordinamento dei valori delle variabili a,c
    if (a > c)
    {
        // Scambio dei valori delle due variabili a,c
        t = a;
        a = c;
        c = t;
    }
    // la variabile a contiene ora sicuramente il
    // valore più piccolo tra quelli inseriti.

    // ordinamento dei valori delle variabili b,c
    if (b > c)
    {
        // Scambio dei valori delle due variabili b,c
        t = b;
        b = c;
        c = t;
    }
    cout << "\n L'ordine voluto e': ";
    cout << a << " " << b << " " << c;

    cout << "\n\nFINE ELABORAZIONE\n";
    system("PAUSE");
    return EXIT_SUCCESS;
} // end main
```