

Vademecum per la codifica dei programmi in C/C++

prof. Mario Barbareschi

Uso del documento: questo documento riporta i riferimenti principali utili per la redazione di programmi in C/C++ per le esercitazioni. Inoltre, è l'unico documento ammesso in seduta d'esame, oltre la traccia fornita dal docente.

Tipi ammissibili

Sono ammessi i seguenti tipi: `char`, `int`, `float` (virgola mobile in singola precisione), `double` (virgola mobile in precisione doppia), `bool`.
Sono altresì ammessi i modificatori di tipo `long` e `short`.

Parole riservate

Non è possibile utilizzare le seguenti parole per nomi di tipi, variabili, funzioni e costanti: `break`, `case`, `char`, `continue`, `default`, `do`, `double`, `else`, `extern`, `float`, `for`, `goto`, `if`, `int`, `long`, `namespace`, `register`, `return`, `short`, `static`, `struct`, `switch`, `typedef`, `union`, `using`, `unsigned`, `while`.

Costanti

Macro: `#define NOME_COSTANTE VALORE_COSTANTE;`
Variabile costante: `const tipo nome = valore;`

Strutture di controllo

Selezione binaria <pre>if(<i>condizione</i>){ //body } else { //se condizione è falsa //body }</pre>	Ciclo a controllo iniziale <pre>while(<i>condizione</i>){ //body }</pre>
Ciclo a controllo finale <pre>do{ //body }while(<i>condizione</i>);</pre>	Ciclo a conteggio <pre>for(<i>inizio</i>; <i>test</i>; <i>incremento</i>){ //body }</pre>

Selezione multipla

```
switch(var){  
    case value0:  
        //body  
    break;  
    case value1:  
        //body  
    break;  
    default:  
        //body  
}
```

Gestione dell'I/O utente

La libreria `iostream` contiene due oggetti dedicati allo stream: `cout` e `cin`.

Uso del `cout`: utilizzare `cout` in combinazione con l'operatore `<<` per combinare una stringa in uscita.

Uso del `cin`: utilizzare `cin` in combinazione con l'operatore `>>` per popolare una variabile del programma. La chiama a `cin` è bloccante!

Uso del `cout`

```
float pi_greco = 3.14;  
int raggio = 3.14;  
cout << "Il PI vale " << pi_greco << " mentre il raggio " << raggio <<  
endl;
```

Uso del `cin`

```
float peso;  
cout << "Inserisci il valore del peso (numero reale): ";  
cin >> peso;  
cout << "Valore inserito: " << peso << endl;
```

Operatori aritmetici e logici

Somma: `+`. Sottrazione: `-`. Moltiplicazione: `*`. Divisione: `/`. Modulo: `%`. And bitwise: `&`. Or bitwise: `|`. Negazione bitwise: `~`. Shift a sinistra: `<<`. Shift a destra: `>>`. And logico: `&&`. Or logico: `||`. Negazione logica: `!`

Utilizzando l'operatore di assegnazione preceduto da una delle operazioni causa l'assegnazione alla variabile sinistra l'operazione applicata tra la variabile a sinistra e l'operando (variabile o costante) a destra.

```
int distanza = 10, tratto = 5;  
distanza += tratto;
```

Operatori di confronto

Gli operatori di confronto possono essere utilizzati per verificare delle condizioni su variabili e costanti. Maggiore e maggiore o uguale: `>` `>=`. Minore e minore o uguale: `<` `<=`. Operatore di eguaglianza e di disuguaglianza: `==` `!=`

Tipi composti

Record

```
struct nome_struct{
    tipo campo1;
    tipo campo2;
};
```

Array

```
int vettore[10]; //vettore di 10 interi
float matrice[4][4]; //matrice di 4x4
elementi
```

Commenti nel codice

Inserire commenti su singola riga adoperando i simboli //, mentre su più righe /* per cominciare una sezione di commento e */ per chiudere.

Funzioni

Prototipo di funzione

```
void stampaCarattere(char);
```

Dichiarazione funzione

```
void stampaCarattere(char c){
    cout << c;
}
```

Funzione con return

```
int somma(int a, int b){
    return a+b;
}
```

Funzione parametro riferimento

```
void somma(int a, int b, int &c){
    c = a+b;
}
```

Esempio completo di Programma

Funzione parametro riferimento

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

int pow(int x, int n);

int main() {
    int base, esp;
    cout << "Inserisci base: ";
    cin >> base;
    cout << "Inserisci esp: ";
    cin >> esp;
    cout << base <<" alla " << esp << " fa " << pow(base, esp) << endl;
    return 0;
}

int pow(int x, int n){

    int i, ris = 1;
    for (i=0; i<n; i++) {
        ris*=x;
    }
    return ris;
}
```