Elementi di C++ di base

Corso di Programmazione 3 - Ingegneria dell'Informazione e dell'Organizzazione
10 ottobre , 2001

Gino Perna

Esempi di semplici programmi in C++

Il programma più semplice consiste semplicemente di una sequenza di istruzioni, senza cicli, scelte o funzioni (eccetto quelle per le operazioni di I/O od operazioni matematiche).

Questi non sono certo programmi interessanti. Praticamente eseguono le stesse operazioni che potrebbero essere fatte su un calcolatore tascabile. Il problema risolto è quindi volutamente semplice in modo che ci si possa concentrare sulla sintassi.

STRUTTURA GENERALE

```
//arith.cpp -- aritmetica in C++
#include <iostream.h>

int main(void)
{
   float x,y;

   cout << “Dammi un numero:”;
   cin >> x;
   cout << “Dammi il secondo numero:”;
   cin >> y;

   cout << “x = “<<x<<“ y = “<<y<<“\n”;
   cout << “x+y = “ << x+y << “\n”;
   cout << “x-y = “ << x-y << “\n”;
   cout << “x*y = “ << x*y << “\n”;
   cout << “x/y = “ << x/y << “\n”;

  return 0;
```

```
}
Compilazione
```

La compilazione sara' effettuata sulla macchina UNIX con il comando

```
g++ -o arith arith.cpp
```

ove arith.cpp e' il nome del programma appena immesso, arith e' il nome dell'eseguibile che verra' generato.

Per lanciare il programma digitare

```
arith
```

#include

Gli `include` servono ad indicare al compilatore quali classi/funzioni dovranno essere usate. (Normalmente si userà la `iostream.h` e la `math.h`).

commenti

I commenti possono essere di due tipi:

- sulla riga: il commento parte dalla sequenza `//` fino a fine riga
- a blocchi: il commento è tutto ciò che è contenuto tra `/* e */`, anche quindi su più righe; attenzione che commenti annidati non sono possibili.

corpo programma

Normalmente tutto ciò che è compreso tra `int main { ..... return 0;}`

il significato del valore 0 ritornato e' interpretato dal sistema operativo. (0 indica normalmente che non ci sono stati errori).

Il programma presentato esegue le quattro operazioni sui due numeri dati dall'utente.

Si tenga presente che la rappresentazione interna dei numeri non e' esatta e quindi possono esserci errori di arrotondamento. Si provi ad esempio con i numero 1.23 ed 1.18

I limiti numerici dei vari tipi di dati sono molto importanti per un corretto utilizzo della RAM e la correttezza dei risultati (Esempio di approccio diverso AUTOCAD-MICROSTATION).

TIPI DI DATI

float e double

Si provi ora a cambiare il programma sostituendo al tipo `float` il tipo `double` e si riprovi l'esempio.

funzioni trigonometriche

```
// esempio con funzione matematica
#include <iostream.h>
#include <math.h>

int main () {
   double pi = 3.1415926;// pi = atan(1)*4;
   for ( double x = 0; x <= 2 * pi; x += (pi / 4) ){
      cout << " sin(" << x << ") = " << sin(x) << "\n";
   }
return 0;
}
```

```
Questo esempio stampa il seno dell'angolo ogni pi/4 di radiante
(Le funzioni trigonometriche hanno l'argomento in radianti!).
```

Si noti che la stampa del seno di pi non e' esattamente zero in quanto si e' commesso un banale errore di inizializzazione della variabile pi. La corretta inizializzazione si ha con l'istruzione commentata o specificando il valore con 18 cifre sigificative.

Cosa succede se dimentico `<math.h>`?

char

Il tipo char e' forlmalmente un tipo derivato, corrispondente normalmente ad un intero.

Esso contiene uno ed un solo carattere.

```
//  Stampa i caratteri dal codice 32 al 127
#include <iostream.h>
void main() {
  for (int i = 32; i < 127; i++) {
    char c = i;
    cout << c << '\t';
    if ((i % 10) == 2)
      cout << '\n';
  }
}
```

In questo modo si stampano tutti i caratteri dal codice ASCII 32 al codice 126, ovvero tutti i caratteri stampabili.

Si noti il codice `if ((i % 10) == 2) cout << '\n';` che permette di mandare A CAPO ogni 10 caratteri stampati.

funzioni semplici

Si scriva un programma che dato un numero ne restituisca il suo cubo (x*x*x)

```
// calcola il cubo di un numero
#include <iostream.h>

double cubo (double x) {
  return x * x * x;
}

void main () {
  double x;
  cout << "x = "; cin >> x;
  int n = (int)x;
  int m = (int)cubo(x);
  cout << "n = " << n << "\n";
  cout << "m = " << m << "\n";
}
```

Si noti il CAST ovvero la forzatura a trasformare un tipo di dato in un altro.

funzioni con piu' argomenti

```
//  funzioni: calcolo della potenza
#include <iostream.h>
#include <math.h>

double potenza (double base, double esponente) {
  double x = exp (esponente * log(base) );
  return x;
}

void main() {
  double base, esponente;
  while (true) {   // infinite loop
    cout << "\n\nscrivi base = 0 per terminare\n";
    cout << "base? "; cin >> base;
    if (base == 0)
      break;
    cout << "esponente? "; cin >> esponente;
    double x = potenza (base, esponente);
    cout << "risultato = " << x;
  }
}
```

Questo programma contiene le seguenti novita':

1. funzione con piu' argomenti;
2. loop infinito;
3. uscita condizionata ;
4. utilizzo di funzioni matematiche di libreria.

Ovviamente e' cura del programmatore (o dell'utente?) non dare condizioni non valide al programma/funzioni.

TIPI DI DATI DERIVATI

Stringhe, vettori e matrici sono tipi di dati derivati in C++.

I vettori sono collezioni di elementi omogenei identificati da un indice (che parte da 0) ai quali si puo' accedere direttamente specificando l'elemento di interesse. Ogni singolo elemento puo' essere quindi utilizzato come semplice variabile.

Attenzione:

1. Il numero di elementi e' deciso in fase di compilazione
2. Il numero di elementi non puo' variare
3. Non esiste controllo sugli indici durante l'esecuzione (eventuali programmi commerciali esterni).
4. Prestare attenzione agli indici (diversi da PASCAL, FORTRAN,BASIC)!!!!!

Inizializzazione di vettori

```
//esempi di inizializzazione di vettori
#include <iostream.h>
void main() {
  double a[5] = {0, 2.0, 4.0, 6.0, 8.0};
  int b[] = {1, 3, 5, 7, 9};
  for (int i = 0; i < 5; i++)
    cout << a[i] << "\t" << b[i] << "\n";
}
```

I vettori possono essere dimensionati automaticamente (vettore b) oppure esplicitamente (vettore a).

funzioni con vettori

```
// Esempio di dichiarazione inizializzazione e stampa vettore
#include <iostream.h>
#include <math.h>
```

```
const double PI = 3.14159265358979323846264338327;
const int N_ELEM = 9;

void sin (double v[], int n) {
  for (int i = 0; i < n; i++)
    v[i] = sin(v[i]);
}

void stampa (double v[], int n) {
  for (int i = 0; i < n; i++)
    cout << v[i] << "\n";
}

void main() {
  double v[N_ELEM];
  for (int i = 0; i < N_ELEM; i++)
    v[i] = PI/4 * i;
  sin (v, N_ELEM);
  stampa (v, N_ELEM);
}
```

Cosa stampa l'esempio?

Stringhe

Le stringhe sono dei vettori di caratteri. La fine della stringa e' determinata dal carattere corrispondento al codice ASCII 0 indicato come '\0'.

```
// attenzione il programma e' ERRATO
#include <iostream.h>
void main() {
  char s[5];
  // s = "ciao";  // NO: " " solo per l'inizializzazione
  s[0] = 'c';
  s[1] = 'i';
  s[2] = 'a';
  s[3] = 'o';
  // senza carattere '\0' a fine stringa
  cout << s;
}
```

Cosa accade?

```
#include <iostream.h>
void main() {
  char s[20];
  cout << "parola (min 7, max 20)? "; cin >> s;
  cout << "seconda lettera: " << s[1] << "\n";
  cout << "terza lettera: " << s[2] << "\n";
  cout << "settima lettera: " << s[6] << "\n";
```

```
}
```

Cosa accade se non rispetto i limiti?

Calcolo della media

```
// calcola la media dei voti di uno studente
#include <iostream.h>
void main() {
  // il vettore contiene al massimo 50 voti
  const int MAX_VOTI = 50;
  double voti[MAX_VOTI];
  int i; // dichiariamo l'indice i prima
         // del for perche' ci servira
         // all'esterno di esso
  cout << "inserisci un numero negativo per terminare\n";
  for (i = 0; i < MAX_VOTI; i++) {
    double x;
    cout << "voto? "; cin >> x;
    if (x < 0)
      break;
    voti[i] = x;
  }
  // i e' il numero dei voti immessi
  // sommiamo tutti i voti
  // utilizzando un nuovo indice j
  double somma = 0;
  for (int j = 0; j < i; j++)
    somma += voti[j];
  double media = somma / i;
  cout << "la media dei voti e': " << media << "\n";
}
```

Ricordarsi di prestare attenzione al fatto che la media deve essere di tipo `double` mentre il vettore `voti` potrebbe essere intero.

Provare a riscrivere il programma creando una funzione che inserisce i dati ed una che calcola la media.