

Tipi semplici 2

Contenuti rieditati delle slide della
Prof. L. Caponetti

Dati numerici

- I linguaggi di programmazione consentono di elaborare dati numerici: interi e reali
- Il linguaggio C fornisce 3 tipi predefiniti:
 - **int** per rappresentare ed elaborare un sottoinsieme dei numeri interi
 - **float** e **double** per rappresentare ed elaborare un sottoinsieme dei numeri reali

Tipo float

- I numeri reali sono rappresentati in modo approssimato, utilizzando un numero finito di byte, **generalmente 4 byte per la precisione semplice e 8 byte per la precisione doppia**
- I domini dei tipi **float** e **double** costituiscono un sottoinsieme dei numeri reali

Tipo float

- Un **float** è rappresentato in memoria in forma esponenziale normalizzata

$$x = \pm m * b^e$$

- e rappresenta l'esponente o caratteristica, un intero con segno
- m rappresenta la mantissa normalizzata, cioè $1/b \leq m \leq 1$

Notazione esponenziale

- Esempio
 - Base: 10
 - Mantissa $0,1 \leq m \leq 1$

$$321.5 \quad \rightarrow \quad +0.3215 * 10^{+3}$$

$$-0.001 \quad \rightarrow \quad -0.1 * 10^{-2}$$

float e double - dominio

- Il dominio dipende dalla rappresentazione adottata
- I valori del dominio possono essere descritti da 2 attributi:
 - **Precisione** – descrive, in notazione decimale, il numero di cifre decimali rappresentabili
 - **Intervallo** – descrive quale sia il più piccolo ed il più grande numero rappresentabile

float e double - dominio

- **Precisione**

- float – precisione semplice – circa 6 cifre decimali
- double – precisione doppia – circa 15 cifre decimali

- **Intervallo**

- float – $[10^{-38}, 10^{38}]$
- double – $[10^{-308}, 10^{308}]$

Operatori

float x float -> float

- + Addizione
 - - Sottrazione
 - * Moltiplicazione
 - / Divisione
-
- Gli operatori + - * / sono **sovraccarichi**: sono utilizzati per indicare operazioni differenti. Ad esempio + indica la somma tra interi ed anche tra reali

Operatori relazionali - predicati

Restituiscono un valore vero o falso

== Uguale

!= Diverso

< Minore

<= Minore uguale

> Maggiore

>= Maggiore uguale

Osservazioni sull'uso dei reali

- La rappresentazione di un numero reale con una variabile di tipo float introduce generalmente una approssimazione
- Ad esempio, il numero 1 potrebbe essere rappresentato come 0.9999999
- Si potrebbero quindi avere degli errori se si eseguono confronti esatti con le variabili reali

Osservazioni sull'uso dei reali

- Siano x e y 2 variabili di tipo float

- Generalmente si ha che

$(x/y)*y$ è diverso da x

- Esempio

- Invece di scrivere $x==y$ si potrebbe scrivere

$(x \leq y + 0.000001) \ \&\& \ (y \leq x + 0.000001)$

Costanti reali

- Le costanti reali sono rappresentate seguendo la notazione scientifica dei numeri reali, cioè
 - La sintassi prevede una parte intera, eventualmente preceduta dal segno, una parte decimale opzionale preceduta dal punto decimale ed un fattore di scala opzionale
- Esempi: +0.1, 1.51, -0.001, 1.0e+2F (e indica la base 10, F indica float)

Compatibilità

- Sia il tipo T1 compatibile con il tipo T2
 - **Un operatore O2 definito per T2 può essere applicato anche ad un dato di tipo T1**
 - **Il risultato è di tipo T2**
- $O2: T2 \times T2 \rightarrow T2$
- $O2: T1 \times T2 \rightarrow T2$
- $O2: T2 \times T1 \rightarrow T2$

Tipo int compatibile con il tipo float

- Il tipo int è compatibile con il tipo float
 - Un operatore del tipo float può essere applicato ad un int ed a un float producendo un risultato float

Operatori con 2 operandi

	2 interi	2 reali	1 intero, 1 reale
+ - * /	int	float	float
%	int	errore	errore

Compatibilità tra gli interi ed i reali

- Gli interi sono compatibili con i reali: se in una espressione aritmetica vi sono dati interi e reali, gli interi sono convertiti automaticamente in reali ed il risultato è reale, **il valore risultante deve necessariamente essere assegnato ad una variabile di tipo reale**

Compatibilità - esempi

- Si consideri il segmento di programma
 $x=4; y=5; n=x+y; n=(n/2)*2;$
 - Se le variabili x, y, n sono di tipo `int` che valore ha n alla fine? 8
 - Modificare l'ultima istruzione in modo che il valore di n non cambi
 $n=(n/2)*2 + n\%2;$

Tipo char

- **Char**, abbreviazione di character, è un tipo semplice, predefinito del linguaggio C
- Una variabile di tipo **char** può assumere un valore tra i seguenti
 - Lettere: a,b,c,.....z A,B,C,....Z
 - Cifre: 0,1,2,.....9
 - Caratteri speciali: . , : \$.....)
 - Caratteri di controllo per la visualizzazione

Tipo char

- **ASCII** – American Standard Codes for Information Interchange – descrive il formato con cui sono rappresentati i caratteri
- Un carattere è rappresentato in memoria in un **byte**
- Tramite il codice **ASCII** ad ogni carattere è associato un numero intero dell'intervallo [0, 255]
- Un carattere è rappresentato come un intero senza segno

Relazione d'ordine

- Nel dominio del tipo **char** è definita una relazione d'ordine tale che risultino ordinate le lettere dell'alfabeto, le cifre decimali ed i caratteri di controllo
- Ad ogni carattere è associato, mediante il codice **ASCII**, un numero intero che rappresenta il numero d'ordine, la posizione, di quel carattere nell'insieme dei caratteri

Numero d'ordine - esempio

Costante carattere	Numero d'ordine
'0', '1','9'	48, 49,57
'A', 'B','Z'	65, 66,90
'a', 'b','z'	97, 98,112

Operatori relazionali

- Ai valori di tipo char si possono applicare i seguenti operatori relazionali

== Uguale

!= Diverso

< Minore

<= Minore uguale

> Maggiore

>= Maggiore uguale

Costanti di tipo char

- Per denotare un valore del dominio del tipo char lo si racchiude tra apici
- Esempi: 'a' '0' 'Z' '.'