

Fondamenti di Informatica L-A

Esercitazione 2

Variabili ed Espressioni: Dichiarazione, Coercizione, Overloading, Overflow, Underflow, Divide by Zero e Casting.

Input-Output: `printf` e `scanf`.

Paolo Torroni

Università degli Studi di Bologna
Laurea in Ingegneria Elettronica e delle Telecomunicazioni

Anno Accademico 2007/2008

△ 1. Dichiarazione di Variabili ▽

- ▶ Si utilizzi il seguente schema di programma C:

```
#include<stdio.h>
main() {
// dichiarazioni
// [<qual>] [<quant>] <tipo> <var> , <var> , ... ;
// ...
// istruzioni
// ...
}
```

- ▶ Si definiscano le seguenti variabili:
 - ▶ una variabile A per valori interi,
 - ▶ con la minima occupazione di memoria possibile,
 - ▶ che dovrà avere nel proprio dominio i valori da -50000 a 30000;
 - ▶ una variabile B per valori interi,
 - ▶ con la minima occupazione di memoria possibile,
 - ▶ che dovrà avere nel proprio dominio i valori da 0 a 2^{15} (ma non valori negativi);
 - ▶ una variabile C di tipo carattere;
 - ▶ una variabile D di tipo double.

△ 2. Dichiarazione di Variabili e Conversioni di Tipo ▽

- ▶ Si assegnino valori iniziali alle variabili nel modo seguente:

A	←	0
B	←	10^4
C	←	77
D	←	10.0

(in linguaggio C, per esprimere il valore 10^4 , si può usare la notazione $1E4$)

- ▶ Si eseguano le seguenti operazioni:
 - ▶ Si divida B per C (risultato in B)
 - ▶ Si sottragga D a C (risultato in C)
 - ▶ Si assegni a C la metà del valore di B
 - ▶ Si sommi D ad A (risultato in A)
- ▶ Si controlli l'esecuzione del programma con in debugger, per seguire l'evoluzione del valore delle variabili.
- ▶ Che valori visualizza il debugger al termine dell'esecuzione?

△ 3. Conversione, Coercizione e Overloading ▽

- ▶ Si assegnino valori iniziali alle variabili nel modo seguente:

A ← 10	B ← 10 ⁴	C ← 77	D ← 10.0
--------	---------------------	--------	----------

- ▶ Si eseguano le seguenti operazioni, **ripristinando i valori iniziali delle variabili dopo ogni istruzione**:
 1. $C = C \div D \times D$
 2. $C = C \div A \times A$
 3. $C = C \times A \div A$
 4. $C = C \times A$ seguita da $C = C \div A$ (2 istruzioni separate)
 5. $D = D \div 3 \times 3$
 6. $A = A \div 3 \times 3$
 7. $A = (B \div (C + 3)) \div D$
 8. $C =$ resto della divisione di D per 2
- ▶ Si controlli l'esecuzione del programma con in debugger, per visualizzare il contenuto delle variabili.
 - ▶ Che valori visualizza il debugger al termine dell'esecuzione di ciascuna istruzione?
 - ▶ Sono quelli che ci si aspetta?
 - ▶ In quali casi non lo sono? Perché accade ciò?

△ 4. Overflow, Underflow e Divide by 0 ▽

- ▶ Siano date le variabili viste prima, più due nuovi float: E ed F. Si assegnino valori iniziali alle variabili nel modo seguente:

A	←	10	B	←	10^4	C	←	77	D, E, F	←	10^{-10}
---	---	----	---	---	--------	---	---	----	---------	---	------------

- ▶ Si eseguano le seguenti operazioni, **ripristinando i valori iniziali delle variabili dopo ogni istruzione**:

1. $C = C \times 2$
2. $B = C \times B$
3. $A = B \times B \times B$
4. $F = D \times D$ seguito da $F = F \div D \div D$
5. $E = E \times E$ seguito da $D = D \div E \div E$
6. $F = A + D$ seguito da $F = F/A$
7. $D = A + D$ seguito da $D = D/A$
8. $D = A + D$ seguito da $A = A/D$ seguito da $B = B/A$
9. $E = A + E$ seguito da $A = A/E$ seguito da $B = B/A$
10. $A = A/(A + E)$ seguito da $B = B/A$

- ▶ Si controlli l'esecuzione del programma con in debugger, per visualizzare il contenuto delle variabili.
- ▶ Si descriva e motivi il risultato di ciascuna istruzione.

△ 5. Casting ▽

- ▶ Si considerino le espressioni dei punti 3 e 4 che davano risultati non corrispondenti all'aritmetica dei numeri interi e reali.
- ▶ Si utilizzi il meccanismo del **casting esplicito** per far sì che non si verifichino problemi di troncamento, tipi degli operandi, overflow, underflow, e divisione per zero
 - ▶ ove possibile (non è detto che si riesca ad evitare il problema)
 - ▶ e non modificando il tipo delle variabili:
 - ▶ A è di tipo int.
 - ▶ B è di tipo unsigned short int;
 - ▶ C è di tipo char;
 - ▶ D è di tipo double;
 - ▶ E, F sono di tipo float.

△ 6. Input-Output: printf e scanf ▽

- ▶ Esistono due *funzioni* in C che servono a leggere valori da input e visualizzare valori in output:
 - ▶ printf (per visualizzare)
 - ▶ scanf (per leggere).
- ▶ La sintassi di queste funzioni è la seguente:
 - ▶ scanf("%d", &A);
 - ▶ Assegnamento a un int A di un intero letto da tastiera.
 - ▶ Se A è un char, viene trattata da int (codice ASCII).
 - ▶ scanf("%c", &C);
 - ▶ Assegnamento alla variabile C di un carattere letto da tastiera.
 - ▶ scanf("%f", &F);
 - ▶ Assegnamento a F di un numero razionale letto da tastiera.
 - ▶ F deve essere una variabile di tipo float o double.
 - ▶ printf("%d",A); printf("%c",C); printf("%f",F);
 - ▶ Visualizzazione a schermo del valore di una variabile intera A o del codice ASCII di un carattere A;
 - ▶ Visualizzazione a schermo di un carattere C;
 - ▶ Visualizzazione a schermo del valore di un float o double F.
- ▶ Si utilizzino le funzioni printf e scanf per inizializzare e visualizzare il valore delle variabili all'inizio e alla fine dei primi 4 punti del lucido "4. Overflow, Underflow e Divide by 0".