



Laboratorio di Programmazione

Laurea in Bioinformatica

Web: <http://www.scienze.univr.it/fol/main?ent=oi&id=28023&lang=it>

Docente: *Carlo Drioli*

Email: *drioli@sci.univr.it*

Lucidi a cura di
Nicola Drago Carlo Drioli Federico Fontana

Lezione 8

Sommario

- Overflow: tipi interi e floating point
- Esercizi

Overflow e underflow

- Un overflow si verifica quando il risultato di un'elaborazione è un valore più grande del massimo rappresentabile con i bit per il tipo in uso.
- Un underflow si verifica quando il risultato di un'elaborazione è un valore più piccolo del minimo rappresentabile con i bit per il tipo in uso.
- In Java overflow e underflow non generano mai eccezioni in fase di esecuzione ma danno luogo a risultati molto diversi da quelli attesi.

Overflow: tipi interi

- Range di valori rappresentabili con i tipi int e long:

int: [- 2^{31} , $2^{31} - 1$] 4 bytes = 32 bits

long: [- 2^{63} , $2^{63} - 1$] 8 bytes = 64 bits

Esempio: $2^{31}-1 = 2.147.483.647$; cosa succede se cerchiamo di rappresentare 2^{31} con 32 bit?

- Java gestisce l'overflow scartando i byte piu' a sinistra tra quelli che non rientrano nel n. di bytes del tipo usato.

Overflow: tipi floating point

- Range di valori rappresentabili con i float e double:

<i>type</i>	<i>Size</i>		<i>Range</i>	<i>Precision</i>
<i>name</i>	<i>bytes</i>	<i>bits</i>	<i>approximate</i>	<i>in decimal digits</i>
float	4	32	$\pm 3.4 \times 10^{38}$	6-7
double	8	64	$\pm 1.8 \times 10^{308}$	15

- Java ha dei valori speciali per rappresentare situazioni di overflow e underflow con tipi floating point:
 1. "Infinity" (" -Infinity") è restituito quando un'operazione produce un valore eccedente il limite superiore (inferiore)
 2. "NaN" ("not a number") è restituito quando un'operazione produce un risultato indefinito. Ad esempio, 0.0 diviso per 0.0 o la radice quadrata di un numero negativo sono operazioni matematicamente indefinite.

Esercizi su overflow

- Si scriva il programma Overflow che:

1) dichiara una variabile intera di nome `valoreInt` e assegna a essa il valore `2147480000`;

2) considera una variabile intera `i`, la quale all'interno di un ciclo è incrementata progressivamente per 3650 volte

3) a ogni iterazione del ciclo, calcola e stampa il valore ottenuto dalla somma `valoreInt + i`

4) attende dall'utente il permesso per procedere al passo 5
[sugg.: si adoperi il metodo boolean `readSiNo(String)` della classe `ConsoleInputManager`]

5) ripete i passi 1,2,3 adoperando stavolta una variabile long di nome `valoreLong` e una variabile `j` sempre di tipo long

Esercizi su overflow

■ Si scriva il programma Overflow2 che:

- 1) dichiara una variabile intera di nome `valoreInt` e assegna a essa il valore 1;
- 2) esegue un ciclo per 35 volte: a ogni iterazione del ciclo, aggiorna `valoreInt` col valore ottenuto dalla moltiplicazione `valoreInt * 2` e stampa il contenuto della variabile così aggiornata
- 3) attende dall'utente il permesso per procedere al passo 4
- 4) dichiara una variabile double di nome `valoreDouble1` e assegna a essa il valore 1;
- 5) esegue un ciclo per 1035 volte: a ogni iterazione del ciclo, aggiorna `valoreDouble1` col valore ottenuto dalla moltiplicazione `valoreDouble1 * 2` e stampa il contenuto della variabile così aggiornata
- 6) attende dall'utente il permesso per procedere al passo 7
- 7) dichiara una variabile double di nome `valoreDouble2` e assegna a essa il valore 1;
- 8) esegue un ciclo per 1075 volte: a ogni iterazione del ciclo, aggiorna `valoreDouble2` col valore ottenuto dalla divisione `valoreDouble2 / 2` e stampa il contenuto della variabile così aggiornata
- 9) attende dall'utente il permesso per procedere al passo 10
- 10) dichiara una variabile double di nome `valoreDouble3` e assegna a essa il valore `valoreDouble1 / valoreDouble2`
- 11) stampa il contenuto di `valoreDouble3`
- 12) attende dall'utente il permesso per procedere al passo 13
- 13) dichiara una variabile double di nome `valoreDouble4` e assegna a essa il valore `valoreDouble1 * valoreDouble2`
- 14) stampa il contenuto di `valoreDouble4`

Esercizi di ricapitolazione

(frazioni e stringhe)

- Chiameremo frazione base la frazione che individua la classe di tutte le frazioni equivalenti ottenute moltiplicando numeratore e denominatore della frazione base per lo stesso valore intero positivo

Adoperando i metodi forniti dalla classe `Frazione` e l'algoritmo di Euclide (cfr. pp. 1 e 72-73 del libro del corso) si scriva una classe `FrazioneBase` contenente un programma il quale, dato un oggetto `Frazione`, estrae la frazione a esso associata e, mediante l'algoritmo di Euclide, costruisce un nuovo oggetto `Frazione` contenente la corrispondente frazione base.

- La classe `MaxPalindroma` contenga un programma che:
 - 1) acquisisce dal file testuale "stringa.txt" una stringa
 - 2) trova la più estesa sottostringa palindroma contenuta nella stringa d'ingresso (si considerino solo le sottostringhe centrate)

Es.: in "12111233212221", la massima sottostringa palindroma è "123"

Es.: in "12111234543212221", la massima sottostringa palindroma è "12345"

3) stampa la sottostringa trovata (nell'esempio, stampa "123" o "12345").