

Laboratorio di Programmazione

Laurea in Bioinformatica

25 gennaio 2010

1 Programmazione strutturata

1.1 Esercizio 1

Si scriva lo pseudocodice di un programma che calcola la media di due numeri reali (si commenti l'esercizio nel caso che i due numeri siano interi).

1.2 Esercizio 2

Si scriva lo pseudocodice di un programma che calcola la media di n valori x_1, \dots, x_n , in cui n è inserito dall'utente in fase di esecuzione del programma. Si ricordi che la media m di n valori x_1, \dots, x_n si calcola mediante la formula

$$m = \frac{1}{n} \sum_{i=1}^n x_i$$

1.3 Esercizio 3

Si scriva lo pseudocodice per calcolare il valore massimo di una sequenza di 10 numeri interi inseriti dall'utente in fase di esecuzione.

1.4 Esercizio 4

Si scriva lo pseudocodice per calcolare la differenza massima tra n numeri interi, con n inserito dall'utente in fase di esecuzione.

2 JAVA: esercizi introduttivi

2.1 Esercizio 1

Si scriva il codice JAVA di un programma che quando eseguito visualizza il messaggio:

Ciao.

Sono un programma scritto in Java.

2.2 Esercizio 2

Si scriva il codice JAVA di un programma che calcola la media di due numeri interi. Si verifichi il funzionamento con le coppie di valori (3,7) e (4,7) e si cerchi una spiegazione per eventuali errori nel calcolo.

2.3 Esercizio 3

Si scriva il codice JAVA di un programma `areaTriangolo` che calcola l'area di un triangolo di base 5 e altezza 10. Si ripeta l'esercizio nel caso di un triangolo di base 5 e altezza 7.

2.4 Esercizio 4

Si modifichino gli esercizi 2 e 3 in modo che l'utente possa inserire in fase di esecuzione i valori (x,y) e (base,altezza).

2.5 Esercizio 5

Si scriva in JAVA un programma che accetta in input un numero intero n e stampa:

1. il resto della divisione per due
2. il valore $2n$ se n è pari, $3n$ se n è dispari

Il problema deve essere risolto senza far uso di costrutti di controllo.

3 JAVA: esercizi introduttivi sulla classe String

3.1 Esercizio 1

Si dica quale sarà la frase risultante dalla stampa mediante istruzione "println" di ciascuna delle seguenti definizioni di stringa:

```
"Le "+22+" sorelle"  
"Le "+"20+2"+" sorelle";  
"Le "+22+4+" sorelle";  
"Le "+22+"4"+" sorelle";  
"Le "+(22+4)+" sorelle";
```

Si verifichi la correttezza delle risposte scrivendo un programma che stampa tali stringhe.

3.2 Esercizio 2

Si scriva un programma che, dopo aver chiesto all'utente di inserire in tre passi successivi il giorno, il mese e l'anno, visualizza la data nel formato gg/mm/aaaa.

3.3 Esercizio 3

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa e due indici (numeri interi), stampa la sottostringa delimitata dai due indici e la sua lunghezza.

3.4 Esercizio 4

Si scriva un programma che, dopo aver chiesto all'utente di inserire due stringhe di cui la seconda è una sottostringa della prima, stampa le lunghezze delle due stringhe e l'indice che individua la posizione della sottostringa nella prima stringa.

3.5 Esercizio 5

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa ed un carattere, stampa l'indice della prima occorrenza di quel carattere nella stringa e stampa una nuova stringa nella quale quel carattere è sostituito dal carattere "X".

4 JAVA: i costrutti di selezione "if/else" e iterazione "while"

4.1 Esercizio 1

Scrivere un programma che chiede all'utente di inserire i valori di due variabili intere var1 e var2 e, in base ad opportuni confronti fra le variabili, stampa la frase "Il valore di var1 è [uguale al | maggiore del | minore del] valore di var2".

4.2 Esercizio 2

Sul prezzo di un prodotto viene praticato lo sconto del 3% se costa meno di 500 euro e del 5% per prezzi superiori a 500 euro. Scrivere un programma che calcoli il prezzo da pagare.

4.3 Esercizio 3

Scrivere un programma che, lette due stringhe inserite dall'utente, indichi se sono uguali o diverse. Nel caso di stringhe diverse l'applicazione dovrà visualizzarle in ordine lessicografico e poi in ordine di lunghezza [es. 3.6 pag 80].

4.4 Esercizio 4

La lunghezza della stringa di caratteri che esprime il numero 12 nella lingua italiana è 6, pari cioè alla metà del numero stesso. Utilizzando la classe Intero del package prog.utili scrivere una applicazione che legga un numero e verifichi se gode di questa proprietà. [Es. 3.7 p. 81]

4.5 Esercizio 5

Scrivere un programma per calcolare il valore massimo di una sequenza di numeri interi inseriti dall'utente in fase di esecuzione. Per gestire l'interazione con l'utente si usi una frase del tipo "Vuoi inserire un altro numero (s/n)?".

4.6 Esercizio 6

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa ed un carattere, stampa gli indici di TUTTE le occorrenze di quel

carattere nella stringa e stampa una nuova stringa nella quale TUTTE le occorrenze di quel carattere sono sostituite dal carattere "X". Si risolva il problema evitando l'uso del metodo "replace" della classe String.

5 JAVA: il costrutto di iterazione "for"

5.1 Esercizio 1

Si scriva il programma RuotaStringa che:

1. acquisisce da standard input una stringa
2. produce una nuova stringa, la quale è una versione ruotata della stringa immessa. La rotazione avviene spostando tutti i simboli in base al valore assunto da un indice di rotazione, con la regola che simboli che uscirebbero dalla stringa rientrano dalla parte opposta:
 - un indice positivo n provoca la rotazione a destra di n posizioni della stringa
 - un indice negativo n provoca la rotazione a sinistra di n posizioni della stringaEsempio: n = 1 trasforma "remo" in "orem"
Esempio: n = -1 trasforma "remo" in "emor"
3. stampa la stringa così ruotata

5.2 Esercizio 2

Si scriva il programma TestPalindroma che verifica se la stringa immessa dall'utente è una parola palindroma. Esempi di parole palindrome sono: "ara", "anna", "otto", "ingegni", etc.

5.3 Esercizio 3

Si scriva il programma TutteLePalindrome che:

1. acquisisce dal file testuale "stringa.txt" una stringa
2. trova tutte le sottostringhe palindrome contenuta nella stringa d'ingresso
Es.: in "1211123212221", la sottostringhe palindrome sono "121", "11", "111", "21112", "12321", etc.
3. stampa le sottostringhe trovate

5.4 Esercizio 4

Si scriva il programma `CercaInHtml` che, dato un file HTML e una parola da cercare, produce un nuovo file in cui tutte le occorrenze della parola sono evidenziate.

Suggerimenti:

1. in HTML, e' possibile evidenziare (ad es. in giallo) un frammento di testo usando il Tag "style" in questo modo:

```
<span style="background-color: #FFFF00">frammento di testo</span>
```

2. per gestire l'output su file usare la classe `FileOutputManager` della libreria `prog`.

6 JAVA: uso di oggetti

6.1 Esercizio 1

Adoperando i metodi forniti dalla classe Frazione (cfr. pp. 72-73 del libro del corso) si scriva un programma java il quale:

1. Accetta da tastiera due coppie di valori interi positivi, creando due oggetti Frazione corrispondenti rispettivamente riferiti dalle variabili f1 e f2
2. Calcola e stampa la differenza tra la frazione più grande e la più piccola e ne salva il riferimento in f3 (N.B.: il risultato sarà per forza non negativo)
3. Calcola e stampa la divisione tra frazione più piccola e la più grande e ne salva il riferimento in f4 (N.B.: il risultato sarà per forza minore di uno)
4. Copia l'oggetto riferito da f4 in un nuovo oggetto Frazione riferito da f5
5. Verifica se gli ultimi due oggetti sono uguali oppure no e stampa quello riferito da f5
6. Verifica se gli ultimi due riferimenti sono uguali oppure no.

6.2 Esercizio 2

Adoperando i metodi forniti dalla classe Frazione si scriva un programma java il quale:

1. Accetta da tastiera due valori interi positivi, creando un oggetto Frazione riferito dalla variabile f1
2. Calcola e stampa il reciproco della frazione appena data e ne salva il riferimento in f2 (N.B.: qual è il reciproco di una frazione a/b ?)
3. Calcola e stampa la moltiplicazione tra le frazioni riferite da f1 e f2 e ne salva il riferimento in f3 (N.B.: quanto deve valere questa moltiplicazione ?)
4. Calcola e stampa la divisione tra le frazioni riferite da f1 e f2 e ne salva il riferimento in f4 (N.B.: quanto deve valere questa divisione ?)
5. Calcola il quadrato della frazione riferita da f1 e ne salva il riferimento in f5
6. Verifica se gli oggetti riferiti da f4 e f5 sono uguali oppure no e stampa quello riferito da f5

7 JAVA: cast implicito ed esplicito, operatori postfissi e prefissi

7.1 Esercizio 1

Si scriva il programma TypeCast1 che:

1. dichiara tre variabili intere i,j, e k con valori rispettivamente 3,4, e 6
2. usando il cast di tipo implicito o esplicito, calcola la media corretta (rappresentata come float) in almeno 3 modi diversi e la media troncata (rappresentata con un intero) in almeno due modi diversi

7.2 Esercizio 2

Si scriva il programma TypeCast2 che:

1. legge da tastiera tre variabili intere i,j, e k
2. assegna la media dei tre numeri a una variabile del tipo strettamente necessario a evitare la perdita di informazioni

[suggerimento: si usi l'operatore % per le valutazioni sul tipo da usare]

7.3 Esercizio 3

Si scriva il programma TypeCastRand che:

1. legge da tastiera il valore di un intero N
2. genera N numeri casuali interi compresi nell'intervallo [5,10]
3. ne calcola la media

[suggerimento: si usi la funzione Math.random() che genera numeri casuali di tipo double nell'intervallo [0,1] con distribuzione uniforme]

7.4 Esercizio 4

Il programma PrePost_num1 sia scritto usando gli operatori pre e postfissi:

1. stampa il risultato di tre operazioni di incremento postfixo in sequenza (i messaggi a video siano del tipo "i=0", "stampa di i++: 0", "stampa di i++: 1", etc);
2. si ripeta la stessa cosa con l'incremento prefisso
3. si valutino l'espressioni i=i++ e i=++i e si stampino a video dei messaggi del tipo ("i=0", "stampa di i=i++: ...", etc);

4. si valutino le espressioni $i==i++$ e $i==++i$ come condizioni del costrutto IF e si stampi un messaggio che dica se sono sempre vere o sempre false

8 JAVA: il tipo char e il costrutto switch

8.1 Esercizio 1

La classe CharCfr contenga un programma che:

1. Dichiarare due variabili `c` e `d` di tipo `char` e ne acquisisce i valori da tastiera;
2. A fronte di opportuni confronti stampa una o più frasi fra le seguenti:
 - ”`c` e `d` contengono lo stesso carattere `<valore di c,d>` (codice Unicode: `<unicode di c,d>`)”
 - ”`c` e `d` contengono caratteri differenti”
 - ”nella codifica Unicode il carattere `<c>` (`<unicode di c>`) precede `<d>` (`<unicode di d>`)”
 - ”nella codifica Unicode il carattere `<c>` (`<unicode di c>`) precede immediatamente `<d>` (`<unicode di d>`)”

8.2 Esercizio 2

Si scriva un'applicazione per la simulazione di una semplice calcolatrice. Il programma deve scrivere il risultato dopo aver ricevuto da tastiera due numeri e il simbolo dell'operazione desiderata.

8.3 Esercizio 3

Si scriva un'applicazione che stampa tutte le lettere minuscole dell'alfabeto facendo uso dell'operatore postfisso di incremento. Nel codice appena scritto si sostituisca all'operatore postfisso l'operatore prefisso e si osservi il risultato.

8.4 Esercizio 4

Si scriva un'applicazione che permette di specificare somme di numeri interi inserendo da tastiera stringhe come ”1+13+6+123”. [sugg.: si ricorra ai metodi della classe `String` che permettono di individuare l'indice di un carattere (+ in questo caso) e di estrarre sottostringhe]

9 JAVA: metodi e campi statici

9.1 Esercizio 1

Si scriva un programma per la risoluzione delle equazioni di secondo grado
[sugg. si usi il metodo `sqrt` della class `Math` per il calcolo della radice quadrata]

9.2 Esercizio 2

Si consideri un cerchio di raggio r il cui centro coincide con l'origine di una coppia di assi cartesiani. Si scriva un programma che:

1. generando una sequenza di N punti a caso di coordinate x e y comprese fra 0 ed r , si calcoli la frequenza dei punti che cadono nel quarto di cerchio di circonferenza (cioè il rapporto fra i punti che cadono nel quarto di circonferenza e il numero totale di punti generati). Per generare valori casuali si usi il metodo statico `Math.random()`.
2. usando il metodo al punto precedente calcolare un'approssimazione di π sapendo che la frequenza dei punti che cadono nel quarto di cerchio di circonferenza costituisce un'approssimazione del rapporto fra l'area A_q del quarto di cerchio e l'area del quadrato di lato r . Si ricordi poi che $e' 4 * A_q = \pi * r^2$.
3. si confronti il valore stimato di π con il valore della costante statica `Math.PI`.

10 Rappresentazioni numeriche

10.1 Esercizio 1

Si scriva un programma Java che converte un numero intero dalla rappresentazione in una base B generica a quella decimale.

Suggerimenti: si usi il tipo stringa per l'input del numero da convertire; per la conversione da char a valore numerico di una cifra si puo usare ad es. il metodo statico `int getNumericValue(char)` della classe `Character`.

10.2 Esercizio 2

Si scriva un programma Java che converte un numero intero dalla rappresentazione decimale a quella in una base B generica.

10.3 Esercizio 3

Si scriva un programma che esegue la somma di due numeri binari in notazione complemento a 2 forniti dall'utente in fase di esecuzione. Il programma fornisca come risultato la rappresentazione decimale degli addendi e le rappresentazioni complemento a 2 e decimale della somma, segnalando possibili overflow.

11 JAVA: array e collezioni

11.1 Esercizio 1

Si scriva un programma che, dopo aver riempito un array di 5 interi con valori forniti dall'utente, lo stampa al contrario (partendo dall'ultimo elemento inserito, fino al primo).

11.2 Esercizio 2

Si scriva un programma che, dopo aver riempito un array di 5 interi con valori forniti dall'utente, copia gli elementi in ordine crescente in un secondo array e lo stampa.

11.3 Esercizio 3

La classe `OrdinaStringhe` contenga un programma che:

1. inizializza un array di stringhe acquisite da tastiera
2. ordina l'array per lunghezza crescente delle stringhe, adoperando una variabile di tipo stringa come contenitore temporaneo
3. stampa l'array così ordinato

11.4 Esercizio 4

La classe OrdinaStringhe2 contenga un programma che:

1. inizializza un array di stringhe acquisite da tastiera
2. ordina gli elementi dell'array per lunghezza crescente delle stringhe in un nuovo array, senza adoperare alcun contenitore temporaneo
3. stampa il nuovo array

[SUGG.: copiare nel secondo array e poi escludere la stringa più corta del primo array via via fino a vuotare il primo array]

12 JAVA: array di array

12.1 Esercizio 1

La classe OpMatrici contenga un programma che:

1. crea due matrici 2x2, la prima con righe [1 2] e [3 4], la seconda con righe [5 6] e [7 8].
2. esegue la visita completa delle due matrici, prima per RIGA e poi per COLONNA, stampandone il risultato.
3. ne esegue il prodotto matriciale. Per verifica, si noti che il risultato corretto e' una matrice 2x2 con righe [19 22] e [43 50].

12.2 Esercizio 2

La classe LetturaVerticale contenga un programma che:

1. legge e salva in un array di stringhe una sequenza di stringhe da input fintantoché non viene immessa la stringa "FINE"
2. colloca le stringhe in una matrice di simboli avente un numero di colonne pari alla lunghezza della stringa minore, in cui la riga n-esima contiene i simboli della stringa n-esima
3. presenta a video una nuova sequenza, formata leggendo la matrice colonna dopo colonna.

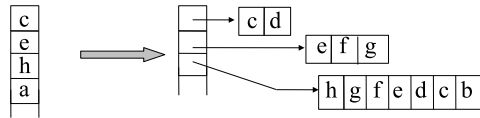
Si risolva il problema prima adoperando un array di array di caratteri, poi lavorando col solo array di stringhe iniziale (ovvero senza creare la matrice di caratteri)

12.3 Esercizio 3

La classe CaratteriNelMezzo contenga un programma che:

1. inizializza un array di 10 simboli UNICODE scelti a caso tra i codici decimali 1 e 150 adoperando il metodo statico random() di Math
2. sostituisce col carattere 'a' gli elementi nell'array che non sono alfabetici minuscoli
3. inserisce carattere 'z' nell'ultimo elemento dell'array, e infine stampa gli elementi dell'array risultante
4. per ogni elemento dell'array escluso l'ultimo, costruisce un nuovo array che contiene tutti i caratteri intermedi tra quello contenuto in quell'elemento (compreso) e quello contenuto nell'elemento di indice superiore

(escluso), in avanti o all'indietro a seconda che l'elemento di indice superiore contenga un carattere rispettivamente posteriore o anteriore nell'ordinamento fissato dal codice UNICODE. Ad esempio:



5. stampa, riga dopo riga, ciascuna lista di caratteri contenuti nei nuovi array così definiti

13 JAVA: Esercizi vari di ricapitolazione

13.1 Esercizio 1

Si scriva un programma in Java che calcola la media di una sequenza di n valori x_1, \dots, x_n , in cui n e i valori della sequenza sono inseriti dall'utente in fase di esecuzione del programma. Si ricordi che la media m di n valori x_1, \dots, x_n si calcola mediante la formula

$$m = \frac{1}{n} \sum_{i=1}^n x_i$$

13.2 Esercizio 2

Si scriva un programma Java che calcola la differenza massima tra n numeri interi, con n e gli n numeri inseriti dall'utente in fase di esecuzione.

13.3 Esercizio 3

Scrivere un programma che, letta una sequenza di frazioni, determini la frazione minore e la frazione maggiore che compaiono nella sequenza, e la stampi a video.

13.4 Esercizio 4

Scrivere un programma OrdinaMaggioSettembre che:

1. acquisisce un testo di una riga dal file maggio_settembre.txt, il quale contiene solo occorrenze delle stringhe "maggio " e "settembre " in sequenza disordinata (notare la presenza dello spazio)
2. ordina tutte le occorrenze lessicograficamente
3. stampa il risultato dell'ordinamento via via che questo procede

[suggerimento: eliminare le parole una alla volta dal testo, e disporle in testa o in coda a una nuova stringa a seconda del loro ordine lessicografico]

13.5 Esercizio 5

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 $\text{valoreInt} + i$
4. attende dall'utente il permesso per procedere al passo 5 [sugg.: si adoperi il metodo booleano `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`

13.6 Esercizio 6

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 $\text{valoreInt} * 2$ 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 $\text{valoreDouble1} * 2$ 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 $\text{valoreDouble2} / 2$ 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 $\text{valoreDouble1} / \text{valoreDouble2}$
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 $\text{valoreDouble1} * \text{valoreDouble2}$
14. stampa il contenuto di `valoreDouble4`

13.7 Esercizio 7

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.

13.8 Esercizio 8

Si scriva un programma che, a fronte di una stringa di lunghezza n inserita dall'utente, generi e stampi la stringa palindroma di lunghezza $2n$ ottenuta dalla stringa inserita.

13.9 Esercizio 9

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").

13.10 Esercizio 10

Si scriva un programma Java che converte un numero frazionario dalla rappresentazione decimale a quella in una base B generica.