

Laboratorio di Programmazione

Laurea in Bioinformatica

28 febbraio 2008

1 Esercizi di ricapitolazione (Java e Perl)

1.1 Esercizio 1

Si progetti una classe di nome `CodiceIntero`, i cui oggetti codificano numeri interi. Ogni oggetto della classe rappresenta un numero intero le cui cifre sono ordinate nel verso opposto a quello del numero con cui l'oggetto è stato creato. Se, ad esempio, il numero in questione è 74235 allora il corrispondente oggetto `CodiceIntero` rappresenta il codice 53247. Dopo avere previsto nella classe le necessarie variabili d'istanza, si progettino i seguenti metodi:

1. (costruttore/codificatore) `public CodiceIntero (int numero)`
2. (decodificatore) `public int decodifica()`
3. (stampa del codice) `public String toString()`

Infine, si dia un metodo `main` che a partire da un numero intero istanzia un oggetto `CodiceIntero` e successivamente stampa il valore del relativo codice e della sua decodifica.

Nota bene: soluzioni che si basino su preesistenti metodi di libreria che effettuano l'inversione di stringhe NON verranno accettate.

1.2 Esercizio 2

Si progetti una classe di nome `Numeri`, i cui oggetti specificano un array `a` di stringhe di dimensione 100. Ciascun elemento dell'array contiene una stringa binaria il cui simbolo iniziale è 1, mentre tutti i restanti simboli sono 0.

Dopo avere previsto nella classe le necessarie variabili d'istanza, si progettino i seguenti metodi:

1. `public Numeri (String numero)`, il quale attraverso l'uso di una procedura ricorsiva (che eventualmente si appoggi su un metodo ausiliario) assegna agli elementi dell'array stringhe ottenute estraendo via via dal parametro numero sottostringhe secondo il formato previsto. Se, ad esempio, numero = "10011000", allora si avrà `a[0]="100"`, `a[1]="1"`, `a[2]="1000"`, `a[3]=null`, . . . , `a[99]=null`.
2. `public String toString()`, il quale restituisce una stringa formata concatenando gli elementi dell'array (esclusi i riferimenti null) attraverso la sequenza di escape `newline`. Nell'esempio precedente, `toString()` restituirà

```
100
1
1000
```

Infine, si dia un metodo `main` che a partire da una stringa binaria obbligatoriamente iniziata dal simbolo 1 istanzia un oggetto `Numeri` e, successivamente, stampa l'oggetto.

Nota bene: soluzioni che si basino su algoritmi di tipo iterativo per la creazione dell'oggetto NON verranno accettate.

1.3 Esercizio 3

Si modifichi la classe `CarteFrancesi` in modo che sia possibile simulare la distribuzione di una mano di poker ad un unico giocatore. Per modellare la singola mano si consiglia di adoperare un ulteriore array di interi `mano[i]` di lunghezza 5.

I metodi aggiuntivi siano:

1. un metodo `public void daiCarte()`, che dà le 5 carte
2. un metodo `public String vediCarte()`, che visualizza le 5 carte in mano
3. un metodo `public void accomodo(int[] cambio)`, che sostituisce le carte nelle posizioni indicate nel vettore `cambio` con altrettante carte prese dal mazzo
4. un metodo `public String combinazione()`, che individua la combinazione finale (coppia, doppia coppia, tris, scala, full, colore, poker, scala reale) .

[SUGGERIMENTO: per individuare la combinazione, è consigliabile ordinare modulo 13 le 5 carte (cioè in base al numero e non al seme) e ricavare un vettore delle differenze. Su questo vettore, si possono facilmente individuare i pattern delle combinazioni.

Es:

```
[#0##] -> coppia      [000#] -> poker
[00##] -> tris         [00#0] -> full
... etc.
```

La classe chiamante, estensione di `UsaCarte`, dà all'unico giocatore 5 carte dopo aver mescolato il mazzo, gli mostra le 5 carte, gli chiede quante e quali carte vuole cambiare e gli comunica infine la combinazione finale individuata.

1.4 Esercizio 4

Sia dato il seguente script Perl, in cui ogni riga è stata numerata per comodità di lettura:

```
1. my $conta = 0;
2. my @elemento=();
3. while (<STDIN>) {
4.     chomp;
5.     last if $_ == 0; #N.B: l'istruzione last interrompe il ciclo
6.     $elemento[$conta++] = $_;
7. }
8.
9. my @arrayfinale=();
10. my $scambia;
11. foreach $i (0..($conta - 2)) {
12.     foreach $j (($i + 1)..($conta - 1)) {
13.         if ($elemento[$j] > $elemento[$i]) {
14.             $scambia = $elemento[$j];
15.             $elemento[$j] = $elemento[$i];
16.             $elemento[$i] = $scambia;
17.         }
18.         elsif($elemento[$j] == $elemento[$i]) {
19.             $elemento[$j]="N";
20.         }
21.     }
22.     (!(($elemento[$i] eq "N")) and @arrayfinale=(@arrayfinale,$elemento[$i]));
23. }
24. print "@elemento\n";
25. print "@arrayfinale";
```

Si descriva sinteticamente (nello spazio lasciato tra una domanda e l'altra):

1. che funzione hanno le prime 7 righe
2. che tipo di algoritmo realizzano le due istruzioni `foreach`

3. qual è il senso dell'inserimento del carattere "N" in *elemento[j]* e il senso dell'istruzione alla riga 22
4. cosa contiene il vettore @arrayfinale al termine dell'esecuzione

Scrivere infine la sequenza di tutti gli assegnamenti al vettore @elemento che avvengono durante l'esecuzione del programma nell'ipotesi che l'utente immetta, in successione, le stringhe 3\n 5\n 7\n 3\n 0\n.

1.5 Esercizio 5

Sia sato il seguente script Perl:

```

1. my @basi = ("A","T","G","C");
2. my $pat = "CGC";
3.
4. my $dim=15;
5. my $conta=0;
6. my @array = ();
7. my $string1="";
8.
9. foreach $j(0..$dim-1){
10.     $array[$j]=$basi[int rand(4)];
11.     $string1=$string1.$array[$j];
12. }
13.
14. print "array : @array \n";
15. print "string1 : $string1 \n";
16.
17. foreach $k(0..$dim-3){
18.     $string2=substr $string1,$k,3;
19.     print "$string2\n";
20.
21.     if($string2 eq $pat){
22.         $conta++;
23.         foreach $m(0..2){
24.             $array[$k+$m]="X";
25.         }
26.     }
27. }
28.
29. print "$conta\n";
30. print "array : @array \n";

```

Si descriva sinteticamente:

1. che funzione hanno le prime 15 righe [N.B.: la funzione int rand(4) genera un numero a caso tra 0 e 3]
2. che tipo di algoritmo realizzano le istruzioni 17-27
3. che cosa conta la variabile \$conta
4. cosa contiene l'array @array al termine dell'esecuzione

Nell'ipotesi che in una esecuzione dello script la stringa @string1 venga ad assumere il seguente valore: ACGCGCCGCTTACTA, si dica quale sarà il valore di \$conta e il contenuto di @array al termine dell'esecuzione.

1.6 Esercizio 6

Sia dato il seguente script Perl:

```

1. my @array1=("CGC","TCG","TAT","CCT");
2.
3. my $conta=0;
4.
5. my $string1=<STDIN>;
6. chomp($string1);
7.
8. my @string2="";
9. my @string3="";
10.
11. my $strlen=length $string1;
12.
13. my $i=0;
14. my $found;
15.
16. while($i lt $strlen-2){
17.     $string2=substr $string1,$i,3;
18.
19.     $found=0;
20.     foreach $j(0..3){
21.         if($string2 eq $array1[$j]){
22.             print "***** $array1[$j]*****\n";
23.             $conta++;
24.             $found=1;
25.         }
26.     }
27.
28.     if($found eq 1){
29.         $string3=$string3."YYY";
30.         $i=$i+3;
31.     }
32.     else{
33.         $string3=$string3."X";
34.         $i=$i+1;
35.     }
36. }
37.
38. print "$conta\n";
39. print "$string3\n";

```

Si descriva sinteticamente:

1. che funzione hanno le prime 14 righe
2. che tipo di algoritmo realizzano le istruzioni 16-36
3. che cosa conta la variabile \$conta
4. cosa contiene la variabile \$string3

Riprodurre poi la successione dei messaggi stampati per effetto delle istruzioni print nell'ipotesi che l'utente immetta la seguente stringa: GAATATCCCGC. Riportare infine i valori di \$conta e \$string3 al termine dell'esecuzione.