Fondamenti di Architetture e Programmazione
II Compitino (Vers. B)

16 dicembre 2005

Cognome .... Nome ....
Matricola ....

Alcune delle seguenti domande riguardano il vostro numero di matricola. Scrivete il vostro numero nel seguente schema, una cifra per ogni spazio:

A	B	C	D	E	F

Nel seguito, useremo le lettere ABC... per riferirci alle corrispondenti cifre del numero di matricola. Ad esempio, "il numero DEF" è il numero costituito dalle ultime tre cifre del vostro numero di matricola.

Rispondete alle seguenti domande[1]:

1. Assumete che x e y siano variabili **int**, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente AF e BE. Dite quale sarà il loro valore *dopo* gli assegnamenti indicati:

[1] Nei programmi, assumeremo sempre che `in` e `out` siano due variabili di classe ConsoleInputManager e ConsoleOutputManager (rispettivamente), già dichiarate e inizializzate.

Assegnamento	x	y
y=++x+1		
x=y==x? x : y		
x=y+(y>x? y : x)		
x=y/5−1		
y=(**int**)(x/5.0)+y++		
x*=y+1		

2. Il seguente programma permette a un utente di introdurre numeri interi, usando il valore speciale 0 per terminare. Alla fine, il programma deve stampare quanti dei valori introdotti non sono multipli di 4 oppure sono multipli di 100. Ad esempio, se l'utente inserisce *60 300 50 25 4 400 0* il programma deve stampare 5 (poiché solo 300, 50, 25, 400 e 0 non sono multipli di 4 oppure sono multipli di 100). Completate le parti mancanti (indicate con dei puntini):

```
int x, ok = 0;

do {

    x = ........................................;

    if ( ........................................ )

        ok++;

} while ( ........................................ );

out.println( ........................... );
```

3. Il seguente programma, dopo aver letto due stringhe, deve stampare quanti caratteri diversi si trovano in posizione uguale. Per esempio, se le stringhe sono *carcassa* e *vasca* deve stampare 2 (**car**cassa e **vas**ca). Completate le parti mancanti (indicate con dei puntini):

```
String s1, s2;

int lung, i, comuni;

s1 = in.............................;
s2 = in.............................;
```

```
lung = Math.min( s1.length(), s2.length() );

comuni = ..............................;

for ( i = 0; i < ..............................; i++ )

    if ( ........................................ )

        comuni..............................;

out.println( ............................ );
```

4. Considerate il seguente programma Java?

```
int x, y, z;
x = in.readInt();
y = in.readInt();
z = 0;
while ( x <= y ) {
  x++;
  z+=2;
}
out.println( z );
```

- Cosa stampa il programma quando $x = EF$ e $y = AB$? ....
- Cosa stampa il programma quando $x = AB$ e $y = EF$? ....

5. Supponete di avere dichiarato e inizializzato le seguenti variabili:

```
int x = DE, y = AB;
long z = AE;
double d = EF.AB;
float f = (float)A.E;
```

Per ciascuno dei seguenti assegnamenti, indicate se l'assegnamento produce un errore di compilazione, e in caso negativo dite che valore viene assegnato alla variabile:

- `x = y + (int)d`: ....
- `d =  y`: ....
- `d = f * x`: ....
- `f = d`: ....
- `d = z + (int)f`: ....

6. Sia data la seguente definizione di variabili:

```java
int x = EF;
int y = DF;
```

Dire qual è il valore di verità delle seguenti espressioni booleane e qual è il contenuto delle variabili x e y dopo la valutazione delle espressioni:

Espressione	**Valore di verità**	x	y
`(y=x++)%2==0 && (++x>y)`			
`(y=x)>x | (x++>y)`			

7. La classe Integer contiene un metodo statico di nome parseInt che ha come solo argomento una String e restituisce un **int** che è il valore della stringa, posto che sia costituita solo da cifre e che rappresenti quindi un intero. Il prototipo del metodo è

   ```java
   public static int parseInt( String s )
   ```

   Supponete che s sia una variabile di tipo String che contiene la rapprsentazione sottoforma di stringa di un intero, e di voler determinare se il suo contenuto è un numero pari, mettendo il valore booleano risultante nella variabile boolean z. Scrivete l'istruzione:

   ....

8. Scrivete un (frammento di) programma Java che operi come segue: legge 100 interi da tastiera e chiede all'utente di inserire un valore x. Quindi, il programma stampa i soli valori inseriti che sono maggiori di x.

   Ecco un esempio di esecuzione (le parti in grassetto sono state inserite dall'utente):

   `Numero 0:` **12**
   `Numero 1:` **1288**
   `Numero 2:` **342**
   `Numero 3:` **342**
   `Numero 4:` **23413**
   `...`
   `Numero 99:` **23**

   `Limite:` **380**

   ```
   1288
   23413
   ...
   ```

9. Scrivete un programma Java che legga una stringa e la stampi al contrario. Per esempio, se la stringa fosse *carico* il programma dovrebbe stampare *ocirac*.