Corso di laurea in Informatica AA 2007/08 Programmazione, M.A. Alberti

Prova scritta – V appello, 23 settembre 2008

Cognome **Nome**

Matricola

1

Nel gioco della tombola vengono estratte una pedina alla volta, senza un preciso ordine, contrassegnata con un numero da 1 a 99, estremi inclusi. Un numero estratto non è più disponibile per il resto del gioco. Il gioco può proseguire fino a quando le pedine non sono finite. Simulate l'estrazione casuale delle pedine della tombola. Occorre definire una struttura dati per memorizzare i numeri già estratti; ad esempio un array le cui posizioni corrispondono ai numeri estraibili e il valore di ciascuna posizione (1 o 0) indica lo stato del numero corrispondente alla posizione nell'array: 1 già uscito e 0 altrimenti.
Definite quindi la classe `Tombola`, che contiene il campo `estratti` ed il metodo `estrai_prossimo()`, che userà la classe Random del pacchetto `java.util`, tiene aggiornato il campo estratti e riporta all'ambiente chiamante il numero estratto. Il metodo `main` consentirà di giocare fino all'estrazione di tutte le pedine e stamperà di volta in volta il numero estratto.

```
import java.util.Random;
public class Tombola{
   final static int TOMBOLA = 99;
   int[] estratti = new int[TOMBOLA +1];

   public int estrai_prossimo() {
      int generato;
      Random rand = new Random();
      do
         generato = Math.abs(rand.nextInt() % TOMBOLA);
      while (estratti[generato] !=0);
      estratti[generato] = 1;
      return generato;
   }

   public static void main (String[] p) {
      final int MAXLINE = 10;
      Tombola tombola = new Tombola();
      int pos = 1;
      for (int i=1; i<TOMBOLA+1; i++) {
         System.out.print(tombola.estrai_prossimo() + "\t");
         if (pos == MAXLINE) {
            System.out.println();
            pos = 1;
         }
         else pos++;
      }
```

```
    }
}
```

2

Supponendo che sia stata dichiarata la variabile `int i`, nel seguente spezzone di codice, stabilire qual'è l'output di ogni ciclo e della variabile che controlla il ciclo alla fine di questo:

`for (i = 1; i < 11; i++) {`
`  if (i%5 == 0) break;`
`  System.out.println (i);`
`}`
 	
`System.out.println ("i: " + i);`
	1
	2
	3
	4
	i: 5

`for (i = 1; i < 11; i++) {`
`  if (i%5 == 0) continue;`
`  System.out.println (i);`
`}`
 	
`System.out.println (("i: " + i);`
	1 8
	2 9
	3 i: 11
	4
	6
	7

`for (i = 1; i < 11; i++) {`
`  if (i%5 == 0) System.exit(0);`
`  System.out.println (i);`
`}`
 	
`System.out.println (("i: " + i);`
	1
	2
	3
	4

3

Data la funzione ricorsiva di una data classe:

```
public static int f(int m, int n) {
   if (m < 5)
      return n;                        /indirizzo A
   else if (m > n)
           return 1 + f(m-1, n+1);     /indirizzo B

        else
            return 1 - f(m-2, n+2);    /indirizzo C

}
```

calcolate il valore riportato dalle funzioni e date una traccia del processo ricorsivo indicando le chiamate successive alla prima e mettendo in evidenza l'indirizzo in cui avviene la chiamata ricorsiva

```
f(7, 1) = 7
traccia dell'esecuzione:
f(7, 1) →
    1 + f(6, 2)            indirizzo B
        1 + f(5, 3)        indirizzo B
            1 + f(4, 4)    indirizzo B
                4          indirizzo A

f(6, 2) = 6
f(6, 2) →
    1 + f(5, 3)            indirizzo B
        1 + f(4, 4)        indirizzo B
            4              indirizzo A
```

```
f(7, 7) = 11
f(7, 7) →
    1 - f(5, 9)          indirizzo C
        1 - f(3, 11)     indirizzo C
            11           indirizzo A
```

4

Stabilire l'output del seguente programma e dire cosa computa:

```
public class Test {
   public static void main (String[] arg) {
      for (int i=1; i<34; i++)
         if (funzione(i)) System.out.println (i);
   }

   static boolean funzione (int n) {
      if (n <= 2) return true;
      if (n%2 == 0) return false;
      for (int i = 3; i < n/2; i += 2)
         if (n%i == 0) return false;
      return true;
   }
}
```

Nel ciclo `for` del metodo `main` si invoca il metodo funzione e si stampa il valore del parametro solo se il metodo ritorna `true`.
Il metodo `funzione` ritorna il valore `true` o `false` in funzione del suo parametro d'ingresso: riporta `true` per `n = 1`, o `n = 2`; `false` se `n` e' pari e negli altri casi si controlla se sia multiplo dei numeri compresi tra 3 e n/2. In altre parole si controlla se il parametro d'ingresso sia primo.
In conclusione lo spezzone stampa i numeri primi compresi tra 1 e 33.

Output:

```
1
2
3
5
7
11
13
17
19
23
29
31
```

5

Definite un ciclo `while` che ad ogni passo incrementa di `1` il valore di una variabile intera, che chiamiamo `num`, inizialmente posta a un valore intero pseudo-casuale, compreso tra `0` e `MAX`, e si ferma quando la variabile raggiunge un multiplo di `MUL`.
Nel codice scrivere l'inizializzazione della variabile di controllo del ciclo. Alla fine del ciclo un contatore opportunamente inizializzato e gestito ad ogni iterazione darà indicazione di quante esecuzioni del ciclo siano state effettuate.

```
public class Ciclo_while {
   static int MAX = 100;
   static int MUL = 7;
   static int num;
   public static void main (String[] s){
```

```
        int passi = 0;
        Random rand = new Random ();
        num = Math.abs(rand.nextInt() % MAX);
        System.out.println(num);
        while (num%MUL != 0) {
              num++;
              passi++;
        }
        System.out.println(num + " passi: "+passi);
  }
```

Dire in generale quante sono **al più** le esecuzioni del ciclo: **6**

Dire quale è il valore della variabile `num` all'uscita dal ciclo quando la variabile fosse stata inizializzata all'ingresso del ciclo con il valore `37`: **42**

E quanti passi del ciclo sono stati eseguiti: **5**
Idem per il valore d'inizializzazione `num = 99`.
Valore di `num`? **105**

Numero di passi del ciclo? **6**

6

Utilizzando la classe `StringTokenizer` implementare un ciclo for necessario per inizializzare l'array di stringhe `frase`. Ad esempio: se in input è data la frase "`sempre caro mi fu quest'ermo colle`" allora l'array `frase` dovrà essere inizializzato a `[sempre, caro, mi, fu, quest'ermo, colle]`.

```
String riga = in.readLine("sempre caro mi fu quest'ermo colle");

StringTokenizer processore_di_riga = new StringTokenizer(riga);

String[] frase = new String[processore_di_riga.countTokens()];

 for (int i = 0; processore_di_riga.hasMoreTokens(); i++)
     frase[i] = processore_di_riga.nextToken();
```

Se `riga` fosse stata inizializzata con l'istruzione
`riga = in.readLine("bella ciao bella ciao bella ciao ciao ciao");`
come sarebbe stato inizializzato l'array `frase` alla fine del ciclo?

frase = [bella, ciao, bella, ciao, bella, ciao, ciao, ciao]