
Cognome**Nome****Matricola**

1

Assumendo la dichiarazione:

```
Random rand = new Random ();
```

Indicare il range dei valori delle seguenti dichiarazioni:

```
rand.nextInt() % 10;
```

```
(int) (Math.random () * 15);
```

Inoltre scrivere un'istruzione per produrre valori pseudo-casuali nell'intervallo:

0 – 10 usando l'oggetto `rand`5 – 25 usando il metodo `random ()` della classe `Math`

2Generare un numero `DIM` di numeri pseudo casuali interi positivi minori di `MAX` e archivarli, se sono pari, in un array che verrà riportato all'ambiente chiamante.

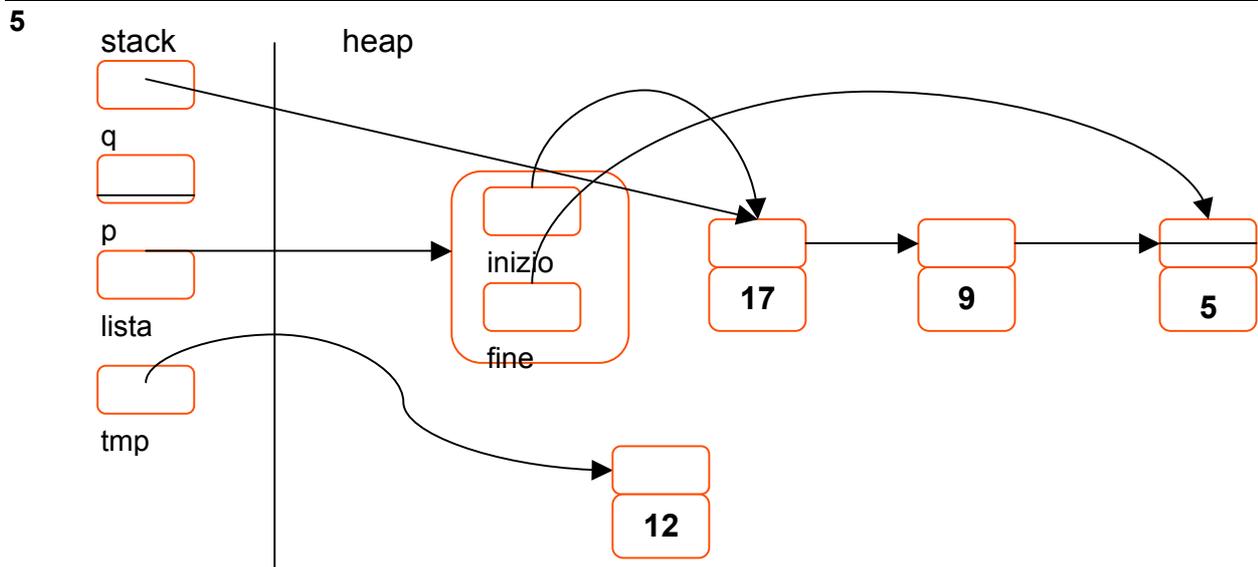
3

Data la seguente funzione ricorsiva:

```
public static int f(int m, int n) {  
    if (m > 5)  
        return n;  
    else if (m > n)  
        return 1 + f(m+2, n-1);  
    else  
        return 1 - f(n+1, m-2);  
}
```

calcolare il valore riportato dalla chiamata `f(4, 3)`:calcolare il valore riportato dalla chiamata `f(2, 7)`:calcolare il valore riportato dalla chiamata `f(3, 1)`:

4Nella classe `Intervallo`, definire il metodo `appartiene(int n)` che riporta un valore booleano per indicare se il numero intero dato appartiene agli insiemi disgiunti `[a, b]` e `[c, d]`, dove i valori `a`, `b`, `c` e `d` sono definiti costanti.



L'immagine rappresenta lo stato dello stack delle chiamate dei metodi e dello heap durante l'esecuzione di un programma che manipola una lista di numeri interi ordinata in senso decrescente, implementata mediante la seguente classe `ListaOrdinata` con la classe interna `NodoLista`:

```
public class ListaOrdinata {
    private NodoLista inizio, fine;

    private static class NodoLista {
        Comparable dato;
        NodoLista pros;
    } .....
}
```

La variabile `lista` sullo stack e' il riferimento alla lista, che nell'immagine viene visualizzata dopo un certo numero di inserimenti. Le due variabili `p` e `q` sono usate come indici di scorrimento della lista per poter effettuare inserimenti e ricerche.

Definite e inizializzate opportunamente, come suggerito dalla immagine, le due variabili `p` e `q`

Scrivete le due istruzioni per far avanzare di una posizione gli indici sulla lista:

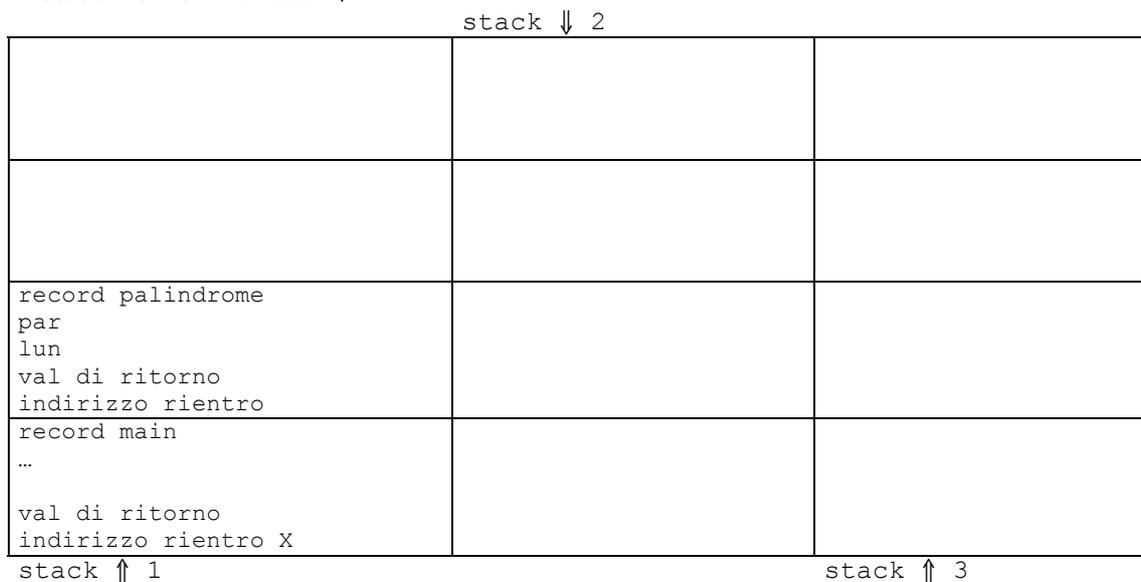
Disegnate inoltre la nuova struttura lista dopo aver inserito il nodo con il dato `12` e lo stato delle variabili al termine dell'operazione d'inserimento.

6

Considerate il metodo ricorsivo `palindrome` che riceve in input una stringa e produce in output un valore di verità per indicare se la stringa è una palindroma (si dice palindroma una parola che può essere letta da sinistra a destra o da destra a sinistra).

```
public static boolean palindrome (String s){
    int lun = s.length();
    if (lun <= 1)
        return true;
    else if (s.charAt(0) != s.charAt(lun-1))
        return false;
    else return palindrome (s.substring(1,lun-1)); // R indirizzo di rientro }
```

Tracciare l'andamento dello stack delle chiamate quando viene invocato `palindrome("esse")` in un metodo `main` all'indirizzo X;

**7**

Per una classe `Tabelline` che implementa una struttura dati a matrice di dimensione 10×10 di interi definite un metodo `calcola()` che riporta la struttura dati a matrice popolata con i dati numerici corretti del calcolo delle tabelline (cioè righe e colonne di un dato indice contengono la tabellina del numero rappresentato dall'indice).

8

In una implementazione dinamica della struttura dati stack, definita come segue

<pre>public class Stack { private NodoStack cima; private class NodoStack { Object dato; NodoStack pros; } public Stack() { cima = null; } }</pre>	<pre>public void push(Object o) { NodoStack t = new NodoStack(); t.dato = o; t.pros = cima; cima = t; }</pre>
---	---

Implementare la funzione `top()` che accede al valore del nodo in cima allo stack e lo riporta all'ambiente senza modificare la struttura e un'eccezione non controllata `EmptyStackException()`.

Si scriva una istruzione `try-catch` in cui si cattura l'eventuale eccezione lanciata dal metodo `top()` nel contesto di un programma in cui sia stata introdotta la variabile `pila`, onde poter recuperare e inserire almeno un nodo. Gli oggetti da inserire nella pila siano di classe `Integer`.

```
Stack pila = new Stack();
```

9

Avendo definito `class Settimana implements java.util.Iterator { }`; dire se sono lecite le seguenti istruzioni:

- a. `Settimana s = new Settimana();`
`while(s.hasNext()) <computa s.next(>` SI NO
- b. `Iterator i = new Settimana();` SI NO
- c. Qual'è il concetto della programmazione a oggetti per cui diversi oggetti che hanno una interfaccia comune rispondono in modo diverso quando un metodo di quell'interfaccia è invocato:
.....
- d. Casting è appropriato quando si riceve un riferimento a una classe antenata e si sappia che l'oggetto è una particolare sotto-classe e si voglia usare l'oggetto nella sua funzionalità completa. SI NO