IV Appello 2006/07 - 11 luglio 2007

Cognome **Nome**

Matricola

1

Definire il prototipo di un metodo `init()` che riporti all'ambiente chiamante un array di interi:

```
int [] init()
```

Definire ora il corpo del metodo, che dovrà popolare un array, di lunghezza assegnata MAX, di numeri pseudo-casuali che siano multipli di 3.

```
public static int[] init () {
   int[] archivio = new int[MAX];
   int numero;
   for (int i=0; i<DIM; i++){
      do
         numero = (int)(Math.random()* MAX);
      while (numero%3 != 0);
      archivio[i] = numero;
   }
   return archivio;
}
```

2

Definite (prototipo e corpo) il metodo `rimuovi` per eliminare da un array passato come parametro, l'elemento della posizione specificata nel secondo parametro. L'array verrà riportato all'ambiente chiamante opportunamente modificato: per rimuovere l'elemento si deve procedere a eseguire lo spostamento di una posizione di tutti i valori dell'array dalla posizione specificata alla fine dell'array. Si supponga cha l'array sia un array di interi.

```
public static int[] rimuovi(int[] a, int p)
{
   for (int i = p; i < a.length-1;)
      a[i]=a[++i];
   a[a.length-1] = 0;
   return a;
}
```

3

Definite la struttura dati `matr` matrice di interi di dimensione `M` x `N`, dove `M` e `N` sono costanti intere date:

```
int[][] matr = new int [M][N];
```

Definite (prototipo e corpo) il metodo `popola` che inizializzi la matrice con i numeri interi successivi da `1` a `N` alla riga 1 e il loro doppio ad ogni riga successiva.

```
public static int[][] init(int[][] m) {
   for (int r = 0, c = 0; c < COLONNE; c++)
      m[r][c] = (r+1)*(c+1);
   for (int r = 1; r < RIGHE; r++)
      for (int c =0; c < COLONNE; c++)
         m[r][c] = 2* m[r-1][c];
   return m;
}
```

4

Data la seguente funzione ricorsiva:

```
public static int f(int m, int n) {
   if (n == 0)
      return 0;
   else if (n > 0)
           return 1 + f(n, m-n);
        else
           return 1 + f(m, m+n);
}
```

calcolare il valore riportato dalla chiamata `f(3, 1)`: **5**

calcolare il valore riportato dalla chiamata `f(-4, 6)`: **8**

calcolare il valore riportato dalla chiamata `f(3, -5)`: **7**

Inoltre dire che cosa rappresenta il numero intero che viene calcolato dalla funzione ricorsiva (in 5 parole comprese gli articoli…)

il numero di chiamate ricorsive

5

Nella classe `Intervallo`, definire il metodo `appartiene(int n)` che riporta un valore booleano per indicare se il numero intero `n`, passato come parametro, sia esterno all'intervallo `[a, b]` ma non interno all'intervallo `[c, d]`, dove i valori `a, b, c` e `d` sono definiti costanti e tali da rendere i due intervalli disgiunti.

```
public static boolean appartiene(int n) {
   return (n <= a || n >= b && n <=c || n >= d);
}
```

6

Completate la figura seguente che rappresenta lo stato dello stack e dello heap quando siano state eseguite le seguenti istruzioni:

```
int num = 17;
int[] vett = new int[3];
Lista lista = new Lista(); lista.aggiungi(Integer(15));
```

La citata classe `Lista` è definita dal codice seguente, dove `NodoLista` è una classe interna:

```
public class Lista {
    private NodoLista inizio;

    private static class NodoLista {
       Comparable dato;
       NodoLista pros;
    }
    public Lista() {
       inizio = null;

 }
```

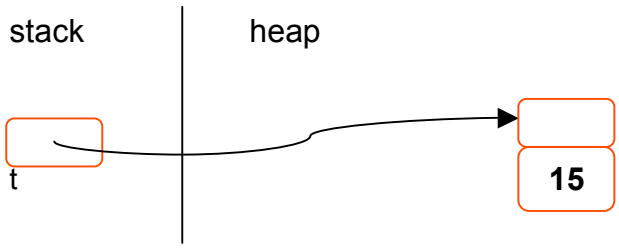
```
public void aggiungi(Object x) {
   NodoLista t = new NodoLista();
   t.dato = x;
   t.pros = null;
   if (inizio == null)
      inizio = t;
   else {
      t.pros = inizio;
      inizio = t;
   }
}
```

7

Dite che tipo di variabile sia la variabile `t` definita nel metodo `aggiungi`

La variabile `t` e' locale al metodo `aggiungi`

Disegnate lo stato dello stack e dello heap relativamente alla sola variabile `t`, al termine dell'esecuzione del metodo `aggiungi` ma prima della sua chiusura, nell'istruzione `lista.aggiungi(Integer(15))`



8

Considerate il codice della classe `Lista` data nell'esercizio precedente e dite come vengono archiviati nella lista i valori numerici.

I numeri da archiviare vengono sempre inseriti all'inizio della lista

Per la stessa classe, definite (prototipo e corpo) il metodo `primoElemento()` che elimina dalla lista il primo nodo e ne riporta il valore archiviato all'ambiente.

```
public Object primoElemento() {
   if (inizio == null)
      throw new CodaVuotaException();
   else {
      Object risultato = inizio.dato;
      inizio = inizio.pros;
      return risultato;
   }
 }
```

9

Considerate i seguenti problemi e dite se sia più utile per ciascun problema una rappresentazione dati statica o dinamica

	rappr. statica (vettore)	rappr. dinamica (lista)
Da un insieme di dati occorre spesso eliminare o inserire un elemento in una data posizione		X
Occorre spesso unire due o più insiemi di dati		X
Occorre spesso leggere un elemento in una posizione nota	X	
Occorre spesso scandire gli elementi di un insieme dati per trattarli	X	
Conosciamo a priori la dimensione dell'insieme dei dati da trattare	X	