

Programmazione I A – Prova scritta del 15 dicembre 2004

Cognome e Nome: _____ Matr.: _____

Esercizio 1. (punti 9) Si consideri un'esecuzione del seguente programma Java (che e' già stato compilato ed eseguito):

```
class StazioneMeteo {
    public String luogo;
    private int[] temperature;

    public StazioneMeteo(String loc, int[] tempe) {
        luogo = loc;
        temperature = tempe;
    }

    public int[] temperature() {return temperature;}
}

class Meteo {

    public static int ricerca(String luogo, StazioneMeteo[] stazioni) {
        for(int i = 0; i < stazioni.length; i++)
            if(luogo.equals(stazioni[i].luogo)) return i;
        return -1;
    }

    static int[] temperatureDi(String luogo, StazioneMeteo[] stazioni) {
        int i = ricerca(luogo, stazioni);
        if(i == -1) return null;
        else return stazioni[i].temperature();
    }

    public static void main(String[] args) {
        int[] tempTorino = {7, 5, 2, 6};
        int[] tempAosta = {3, -2, 0};
        StazioneMeteo[] stazioni = {new StazioneMeteo("Torino", tempTorino),
                                    new StazioneMeteo("Aosta", tempAosta)};

        int[] temp= temperatureDi("Aosta", stazioni);
        System.out.println(temp[1]);
        tempAosta[1] = -5;
        System.out.println(temperatureDi("Aosta", stazioni)[1]);
    }
}
```

1.1 Si raffiguri lo stato della memoria durante l'esecuzione della prima chiamata del metodo `ricerca`; il disegno deve raffigurare lo stato della memoria nell'istante immediatamente precedente la deallocazione del frame di `ricerca`.

1.2 Si scriva l'output prodotto sullo schermo.

Esercizio 2. (punti 10)

- A) Scrivere un metodo *eIn* con tre parametri: un intero *x*, un array *vet* di interi parzialmente riempito con *NumEl* elementi, terzo parametro. *eIn* restituisce *true* se l'elemento *x* compare in *vet*, *false* altrimenti.
- B) Scrivere un metodo *singoli* che richiama *eIn* ed ha due parametri: un array **vettInt** di interi non ordinato, parzialmente riempito, ed un intero **numEl** numero di elementi significativi in **vettInt**. Il metodo *singoli* deve eliminare in **vettInt** gli elementi ripetuti compattando l'array e restituire il numero di elementi significativi dopo le eventuali cancellazioni.
- C) Contare il numero massimo di operazioni che vengono eseguite chiamando *singoli*.

Non si deve far uso di vettori ausiliari e si deve ottimizzare il numero di operazioni.

Suggerimento: il metodo *eIn* dovrà avere un solo ciclo, meglio *while*, da cui uscire non appena si trova uno dei *numEl* di *vet* uguale ad *x*. Il metodo *Singoli* dovrà avere un solo ciclo, meglio *for*.

Esercizio 3. (punti 9)

Si completi il seguente metodo *ordinaIns* che è una specifica dell'algoritmo di *ordinamento (non decrescente) per inserimento*. In *ordinaIns* si usa un secondo metodo *inserOrd*:

```
1. public static void inserOrd(int x, int[] a, int numEl)
2. { int j = 0;
3.   while (j < numEl && a[j] <= x) j++;
4.   for(int i=numEl-1; i>=j; i--)
5.     { a[i+1] = a[i];}
6.   a[j] = x;
7. }

8. public static void ordinaIns(int[] a)
9. { int n = a.length;
10.  for(int i = 1; i < n; i++)
11.  { //Invariante ciclo: a[0...i-1] è ordinato
12.    ..... // completare qui con la chiamata di inserOrd (1 sola istruz.)
    }
  }
```

Rispondere alle domande seguenti motivando le risposte con, dove necessario, riferimenti alle linee di codice della specifica scritta sopra:

- A) Quanti, al massimo, confronti tra elementi del vettore *a* ed *x* esegue il metodo *inserOrd* quando il vettore considerato è già ordinato (non decrescente) e quanti, al massimo, spostamenti sono eseguiti con l'istruzione *a[i+1]=a[i]*?
- B) per il metodo *ordinaIns* combinato con *inserOrd* vi sono un caso peggiore e un caso migliore, cioè casi che richiedono numero di operazioni massimo e minimo? se sì, dire quali sono e dire quale è la loro complessità, cioè il numero di operazioni che vengono eseguite. Tenere conto del punto A.
- C) Dire in cosa la specifica presentata qui differisce da quella presentata nel corso, quale è da preferire e perché'.