

Studente (Cognome Nome): _____

Matricola: _____

Corso di Informatica
Corso di Laurea in Ingegneria Gestionale
a.a. 2005-06
Terzo Scritto – 7 Febbraio 2006

Si noti che le soluzioni ai quesiti saranno considerate valide solo se il materiale consegnato includerà anche lo svolgimento. Tale foglio deve essere consegnato insieme allo svolgimento.

Quesito 1

Un campionato ha 3 squadre, rappresentate da un array di stringhe contenente i nomi delle squadre e da un array di int contenenti i relativi punteggi. Una partita viene rappresentata da un array di 2 stringhe, contenenti i nomi delle squadre, ed un array di 2 int contenenti le reti segnate.

Si dispone inoltre di un metodo

```
public static void ordina_classifica(String[] nomi, int[] punteggi)
```

che, dati un array di stringhe rappresentanti i nomi delle squadre ed un array di int rappresentanti i relativi punteggi, li ordina in modo crescente rispetto ai punteggi.

a) Scrivere un metodo Java che, dati un array di stringhe rappresentanti i nomi delle squadre in classifica, un array di int rappresentati i relativi punteggi, un array di stringhe rappresentanti 2 squadre che si sono affrontate in una partita ed un array di int rappresentati le reti segnate in quella partita, aggiorni la classifica secondo i punteggi acquisiti dalle due squadre con la partita giocata. Il metodo deve avere il seguente prototipo:

```
public static void agg_class(String[] nomi, int[] punteggi, String[] squadre, int[] reti)
```

Inoltre il metodo deve ordinare la classifica in modo crescente. Per tale ordinamento, si può utilizzare il metodo `ordina_classifica` specificato precedentemente, di cui non occorre scrivere il codice.

Ad esempio, nel caso che le squadre siano tre, se `nomi = {inter, roma, livorno}`, `punteggi = {4, 2, 1}`, `squadre = {roma, livorno}`, `reti = {1,2}` (ossia roma-livorno 1-2), dopo l'invocazione del metodo il contenuto di `nomi` sarà `{inter, livorno, roma}` ed il contenuto di `punteggi` sarà `{4,4,1}`. Si ricorda che una vittoria vale tre punti, un pareggio vale un punto per entrambe le squadre, una sconfitta 0 punti.

b) Scrivere un main di test che crei una classifica con i nomi ed i punteggi specificati nel punto precedente, aggiorni la classifica con la partita roma-livorno 1-2, e stampi la classifica in modo decrescente.

Quesito 2

a) Si rappresenti il diagramma di flusso relativo al seguente metodo Java:

```
public static void f(int []a){
    int i;
    for (i=0;i<a.length;i++)
        while (a[i] > 0)
            a[i]-=5;
}
```

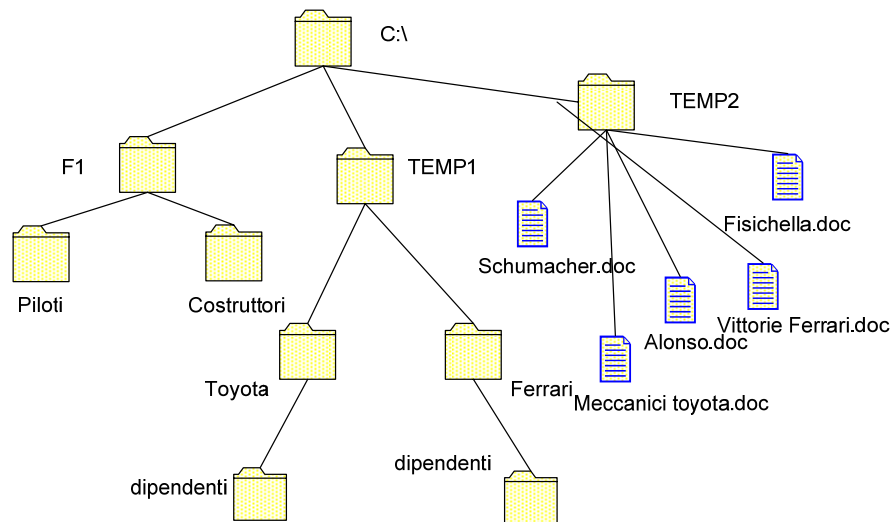
Quesito 3

Si hanno i seguenti numeri: A=511, B=32, C=1026.

- discutere la rappresentabilità su 10 bit in base 2 ed in complemento a due dei numeri sopra indicati. Rappresentare in C2 quelli rappresentabili.
- Discutere la rappresentabilità su 10 bit in C2 delle somme B-A e A-B (B-A va interpretato come B+(-A)). Per quelle rappresentabili, calcolarne la somma.

Quesito 4

Si consideri il seguente file system:



- Impartire la sequenza di comandi per spostare i file e le directory da C:\TEMP1 e C:\TEMP2 nelle relative cartelle/directory, eventualmente creando link. Si suppone che la directory corrente sia C:\. È possibile navigare fra le directory utilizzando il comando cd. Si assume che Schumacher sia un pilota della Ferrari, mentre Fisichella e Alonso siano piloti della Toyota.

Risoluzione Java

```
/*
```

```
* Created on 18-dic-2006
```

```
*/
```

```
/**
```

```
* @author piero
```

```
*
```

```
* To change the template for this generated type comment go to Window -
```

* Preferences - Java - Code Generation - Code and Comments

*/

// implementazione tramite array paralleli senza classi

```
public class arrayparalleli {
```

```
    // metodi di supporto all'ordinamento
```

```
    // precondizioni: gli array sono non null
```

```
    // e gli indici sono validi
```

```
    public static void scambia(String[] S, int i, int j) {
```

```
        String tmp = S[i];
```

```
        S[i] = S[j];
```

```
        S[j] = tmp;
```

```
    }
```

```
    public static void scambia(int[] S, int i, int j) {
```

```
        int tmp = S[i];
```

```
        S[i] = S[j];
```

```
        S[j] = tmp;
```

```
    }
```

```
    // metodo non richiesto: posso definire tutti i metodi che voglio
```

```
    // precondizioni: gli array sono non null
```

```
    public static void ordina_classifica(String[] nomi, int[] punteggi) {
```

```
        int i;
```

```
        int j; // contatori
```

```
        int min, posizione_min; // valore minimo e sua posizione
```

```
        for (i = 0; i < punteggi.length - 1; i++) {
```

```
            min = punteggi[i];
```

```
            posizione_min = i;
```

```
            for (j = i + 1; j < punteggi.length; j++) {
```

```
                if (min > punteggi[j]) {
```

```
                    min = punteggi[j];
```

```
                    posizione_min = j;
```

```

        }
    }
    scambia(punteggi, i, posizione_min);
    scambia(nomi, i, posizione_min);
}
}

public static void agg_class(
    String[] nomi,
    int[] punteggi,
    String[] squadre,
    int[] reti) {
    // si costruisce un array di punti
    // che rappresentano i punti acquisiti dalle squadre con le partite
    int[] punti = new int[2];
    if (reti[0] > reti[1]) {
        punti[0] = 3;
        punti[1] = 0;
    } else if (reti[0] == reti[1]) {
        punti[0] = 1;
        punti[1] = 1;
    } else {
        punti[0] = 0;
        punti[1] = 3;
    }

    // cerca la prima squadra in classifica
    int i;
    for (i = 0; i < nomi.length; i++)
        if (nomi[i].equals(squadre[0]))
            punteggi[i] += punti[0];
        else if (nomi[i].equals(squadre[1]))
            punteggi[i] += punti[1];
}

```

```
        ordina_classifica(nomi, punteggi);
    }
    public static void main(String[] args) {
        String[] nomi = new String[] { "inter", "roma", "livorno" };
        int[] punteggi = new int[] { 4, 2, 1 };

        String[] squadre = new String[] { "roma", "livorno" };
        int[] reti = new int[] { 1, 2 };

        agg_class(nomi, punteggi, squadre, reti);
        // poichè la classifica è ordinata in modo
        // crescente, per stamparla in modo decrescente si stampa al
        // contrario
        int i;
        for (i = 0; i < nomi.length; i++)
            System.out.println(nomi[i] + "\t" + punteggi[i]);
    }
}
```