

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

# Programmazione 1

## A.A. 2013/2014

### Appello del 17 Dicembre 2013 – Compito n° 1

#### Prima parte

#### Esercizio 1 (12 punti)

1.1 Cosa stampa il seguente frammento di codice Java?

```
int [] A = {3, -78, -90, 200, -15, -17, 8};
int j, i = 2;
for (j=0; j<i; j++) {
    if (A[j]>0) i=i+1;
}
System.out.println(A[j]);
```

1.2 Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(1));
    System.out.println(enigma(2));
    System.out.println(enigma(1000));
    System.out.println(enigma(1057));
}
public static int enigma (int x){
    if (x==0) return 0;
    return 3 - enigma (x-1);
}
}
```

#### Esercizio 2 (14 punti)

Un array si dice **equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre equilibrati.

Ad esempio, l'array [1, 1, 10, 4, 6, 2] è equilibrato in quanto  $1+1+10 = 4+6+2$ .

Scrivere un metodo iterativo

**public static boolean equilibrato (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è equilibrato.

#### Esercizio 3 (8 punti)

Un array si dice **totalmente equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà, ed inoltre anche il sottoarray costituito dalla sua prima metà e quello costituito dalla sua seconda metà sono totalmente equilibrati (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre totalmente equilibrati.

Ad esempio, l'array [1, 3, 1, 2, 1, 2] è totalmente equilibrato in quanto  $1+3+1 = 2+1+2$ , ed inoltre anche i sottoarray [1, 3, 1] e [2, 1, 2] sono totalmente equilibrati poiché  $1=1$  e  $2=2$ , rispettivamente; infine, i sottoarray [1], [1], [2] e [2] sono banalmente totalmente equilibrati poiché composti da un solo elemento.

Scrivere un metodo ricorsivo

**public static boolean totalmenteEquilibrato (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è totalmente equilibrato.

## Seconda parte

### Esercizio 4 (16 punti)

Si consideri il seguente tipo di dati:

```
class Riga {
    int lunghezza;
    int somma;
}
```

Scrivere un metodo

**public static Riga[] analizza(int[][] a)**

che preso come parametro un array bidimensionale **a** con  $n$  righe di numeri interi (in cui ogni riga non è necessariamente della stessa lunghezza), restituisce un array di Riga di lunghezza uguale a  $n$ , e tale che l'elemento in posizione  $i$  ( $i=0,\dots,n-1$ ) contiene

- nel campo `lunghezza` la lunghezza dell'array (monodimensionale) che costituisce la riga  $i$  di **a**, e
- nel campo `somma` la somma degli elementi di tale riga.

Ad esempio, se **a** = **{{-10, 35}, {-20}, {-14, 2, 19}}** il metodo deve restituire un array di lunghezza 3 in cui

- in posizione 0 c'è un'istanza della classe `Riga` con `lunghezza=2` e `somma=25`,
- in posizione 1 c'è un'istanza della classe `Riga` con `lunghezza=1` e `somma=-20`,
- in posizione 2 c'è un'istanza della classe `Riga` con `lunghezza=3` e `somma=7`.

### Esercizio 5 [non è possibile utilizzare nessun metodo scritto a lezione senza riscriverlo completamente]

Si consideri il seguente tipo di dati visto a lezione, che rappresenta una lista di numeri interi, identificata dal suo primo elemento.

```
class Elem {
    int valore;
    Elem next;
}
```

#### 5.1 (6 punti)

- Scrivere un metodo iterativo

**public static int massimaDifferenza (Elem lista)**

che presa come parametro una lista di interi, restituisce la differenza tra il massimo ed il minimo valore in essa contenuti. Se la lista è vuota, viene restituito 0.

#### 5.2 (6 punti)

- Scrivere un metodo ricorsivo

**public static int lunghezza (Elem lista)**

che presa come parametro una lista di interi, restituisce la sua lunghezza.

Una lista si dice **sbilanciata a sinistra** se la somma degli elementi nella sua prima metà è maggiore della somma degli elementi nella sua seconda metà (qualora la lista abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Una lista vuota o una lista con un solo elemento sono sempre sbilanciate a sinistra.

Ad esempio, la lista  $1 \rightarrow 3 \rightarrow 14 \rightarrow 4 \rightarrow 6 \rightarrow 2$  è sbilanciata a sinistra in quanto  $1+3+14 > 4+6+2$ .

#### 5.3 (6 punti)

- Scrivere un metodo iterativo

**public static boolean sbilanciataSx (Elem lista)**

che, presa come parametro una lista di numeri interi, restituisce **true** se e solo se **lista** è sbilanciata a sinistra.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **60** minuti per ogni parte.
- **Chi svolge la seconda prova parziale deve consegnare dopo 60 minuti.**
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato e depositato in vista sulla cattedra**
- Mettere in vista sul banco il proprio libretto o altro documento di identità.

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

# Programmazione 1

## A.A. 2013/2014

### Appello del 17 Dicembre 2013 – Compito n° 2

#### Prima parte

#### Esercizio 1 (12 punti)

1.1 Cosa stampa il seguente frammento di codice Java?

```
int [] A = {3, -78, -90, 200, -15, -17, 8};
int j, i = 3;
for (j=0; j<i; j++) {
    if (A[j]>0) i=i+1;
}
System.out.println(A[j]);
```

1.2 Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(1));
    System.out.println(enigma(2));
    System.out.println(enigma(1059));
    System.out.println(enigma(2058));
}
public static int enigma (int x){
    if (x==0) return 1;
    return 4 - enigma (x-1);
}
}
```

#### Esercizio 2 (14 punti)

Un array si dice **sbilanciato a sinistra** se la somma degli elementi nella sua prima metà è maggiore della somma degli elementi nella sua seconda metà (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre sbilanciati a sinistra.

Ad esempio, l'array [1, 1, 10, 4, 6, -2] è sbilanciato a sinistra in quanto  $1+1+10 > 4+6+(-2)$ .

Scrivere un metodo iterativo

**public static boolean sbilanciatoSx (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è sbilanciato a sinistra.

#### Esercizio 3 (8 punti)

Un array si dice **totalmente sbilanciato a sinistra** se la somma degli elementi nella sua prima metà è maggiore della somma degli elementi nella sua seconda metà, ed inoltre anche il sottoarray costituito dalla sua prima metà e quello costituito dalla sua seconda metà sono totalmente sbilanciati a sinistra (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre totalmente sbilanciati a sinistra.

Ad esempio, l'array [3, 3, 1, 3, 1, 2] è totalmente sbilanciato a sinistra in quanto  $3+3+1 > 3+1+2$ , ed inoltre anche i sottoarray [3, 3, 1] e [3, 1, 2] sono totalmente sbilanciati a sinistra poiché  $3>1$  e  $3>2$ , rispettivamente; infine, i sottoarray [3], [1], [3] e [2] sono banalmente totalmente sbilanciati a sinistra poiché composti da un solo elemento.

Scrivere un metodo ricorsivo

**public static boolean totalmenteSbilanciatoSx (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è totalmente sbilanciato a sinistra.

## Seconda parte

### Esercizio 4 (16 punti)

Si consideri il seguente tipo di dati:

```
class Riga {
    int lunghezza;
    int massimo;
}
```

Scrivere un metodo

**public static Riga[] analizza(int[][] a)**

che preso come parametro un array bidimensionale **a** con **n** righe di numeri interi (in cui ogni riga non è necessariamente della stessa lunghezza), restituisce un array di Riga di lunghezza uguale a **n**, e tale che l'elemento in posizione **i** ( $i=0,\dots,n-1$ ) contiene

- nel campo **lunghezza** la lunghezza dell'array (monodimensionale) che costituisce la riga **i** di **a**, e
- nel campo **massimo** il massimo elemento presente in tale riga.

Ad esempio, se **a = {{-10, 35}, {-20}, {50, -30}, {-14, 2, 19}}** il metodo deve restituire un array di lunghezza 4 in cui

- in posizione 0 c'è un'istanza della classe **Riga** con **lunghezza=2** e **massimo=35**,
- in posizione 1 c'è un'istanza della classe **Riga** con **lunghezza=1** e **massimo=-20**,
- in posizione 2 c'è un'istanza della classe **Riga** con **lunghezza=2** e **massimo=50**,
- in posizione 3 c'è un'istanza della classe **Riga** con **lunghezza=3** e **massimo=19**.

### Esercizio 5 [non è possibile utilizzare nessun metodo scritto a lezione senza riscriverlo completamente]

Si consideri il seguente tipo di dati visto a lezione, che rappresenta una lista di numeri interi, identificata dal suo primo elemento.

```
class Elem {
    int valore;
    Elem next;
}
```

#### 5.1 (6 punti)

- Scrivere un metodo iterativo

**public static int massimaDifferenza (Elem lista)**

che presa come parametro una lista di interi, restituisce la differenza tra il massimo ed il minimo valore in essa contenuti. Se la lista è vuota, viene restituito 0.

#### 5.2 (6 punti)

- Scrivere un metodo ricorsivo

**public static int lunghezza (Elem lista)**

che presa come parametro una lista di interi, restituisce la sua lunghezza.

Una lista si dice **equilibrata** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà (qualora la lista abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Una lista vuota o una lista con un solo elemento sono sempre equilibrate.

Ad esempio, la lista  $1 \rightarrow -3 \rightarrow 14 \rightarrow 4 \rightarrow 6 \rightarrow 2$  è equilibrata in quanto  $1+(-3)+14 = 4+6+2$ .

#### 5.3 (6 punti)

- Scrivere un metodo iterativo

**public static boolean equilibrata (Elem lista)**

che, presa come parametro una lista di numeri interi, restituisce **true** se e solo se **lista** è equilibrata.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **60** minuti per ogni parte.
- **Chi svolge la seconda prova parziale deve consegnare dopo 60 minuti.**
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco il proprio libretto o altro documento di identità.

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

# Programmazione 1

## A.A. 2013/2014

### Appello del 17 Dicembre 2013 – Compito n° 3

#### Prima parte

#### Esercizio 1 (12 punti)

1.1 Cosa stampa il seguente frammento di codice Java?

```
int [] A = {3, -78, -90, 200, -15, -17, 8};
int j, i = 1;
for (j=0; j<i; j++) {
    if (A[j]>0) i=i+1;
}
System.out.println(A[j]);
```

1.2 Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(2));
    System.out.println(enigma(1));
    System.out.println(enigma(800));
    System.out.println(enigma(999));
}
public static int enigma (int x){
    if (x==0) return 1;
    return 5 - enigma (x-1);
}
}
```

#### Esercizio 2 (14 punti)

Un array si dice **sbilanciato a destra** se la somma degli elementi nella sua prima metà è minore della somma degli elementi nella sua seconda metà (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre sbilanciati a destra.

Ad esempio, l'array [1, 1, -10, 4, 6, -2] è sbilanciato a destra in quanto  $1+1+(-10) < 4+6+(-2)$ .

Scrivere un metodo iterativo

**public static boolean sbilanciatoDx (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è sbilanciato a destra.

#### Esercizio 3 (8 punti)

Un array si dice **totalmente sbilanciato a destra** se la somma degli elementi nella sua prima metà è minore della somma degli elementi nella sua seconda metà, ed inoltre anche il sottoarray costituito dalla sua prima metà e quello costituito dalla sua seconda metà sono totalmente sbilanciati a destra (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre totalmente sbilanciati a destra.

Ad esempio, l'array [2, 1, 3, 1, 3, 3] è totalmente sbilanciato a destra in quanto  $2+1+3 < 1+3+3$ , ed inoltre anche i sottoarray [2, 1, 3] e [1, 3, 3] sono totalmente sbilanciati a destra poiché  $2 < 3$  e  $1 < 3$ , rispettivamente; infine, i sottoarray [2], [3], [1] e [3] sono banalmente totalmente sbilanciati a destra poiché composti da un solo elemento.

Scrivere un metodo ricorsivo

**public static boolean totalmenteSbilanciatoDx (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è totalmente sbilanciato a destra.

## Seconda parte

### Esercizio 4 (16 punti)

Si consideri il seguente tipo di dati:

```
class Riga {
    int lunghezza;
    int minimo;
}
```

Scrivere un metodo

**public static Riga[] analizza(int[][] a)**

che preso come parametro un array bidimensionale **a** con  $n$  righe di numeri interi (in cui ogni riga non è necessariamente della stessa lunghezza), restituisce un array di Riga di lunghezza uguale a  $n$ , e tale che l'elemento in posizione  $i$  ( $i=0, \dots, n-1$ ) contiene

- nel campo `lunghezza` la lunghezza dell'array (monodimensionale) che costituisce la riga  $i$  di **a**, e
- nel campo `minimo` il minimo elemento presente in tale riga.

Ad esempio, se **a** = **{{-10, 35}, {-20}, {50, -30}, {-14, 2, 19}}** il metodo deve restituire un array di lunghezza 4 in cui

- in posizione 0 c'è un'istanza della classe `Riga` con `lunghezza=2` e `minimo=-10`,
- in posizione 1 c'è un'istanza della classe `Riga` con `lunghezza=1` e `minimo=-20`,
- in posizione 2 c'è un'istanza della classe `Riga` con `lunghezza=2` e `minimo=-30`,
- in posizione 3 c'è un'istanza della classe `Riga` con `lunghezza=3` e `minimo=-14`.

### Esercizio 5 [non è possibile utilizzare nessun metodo scritto a lezione senza riscriverlo completamente]

Si consideri il seguente tipo di dati visto a lezione, che rappresenta una lista di numeri interi, identificata dal suo primo elemento.

```
class Elem {
    int valore;
    Elem next;
}
```

#### 5.1 (6 punti)

- Scrivere un metodo iterativo  
**public static int massimaDifferenza (Elem lista)**  
che presa come parametro una lista di interi, restituisce la differenza tra il massimo ed il minimo valore in essa contenuti. Se la lista è vuota, viene restituito 0.

#### 5.2 (6 punti)

- Scrivere un metodo ricorsivo  
**public static int lunghezza (Elem lista)**  
che presa come parametro una lista di interi, restituisce la sua lunghezza.

Una lista si dice **equilibrata** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà (qualora la lista abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Una lista vuota o una lista con un solo elemento sono sempre equilibrate.

Ad esempio, la lista  $1 \rightarrow -3 \rightarrow 14 \rightarrow 4 \rightarrow 6 \rightarrow 2$  è equilibrata in quanto  $1+(-3)+14 = 4+6+2$ .

#### 5.3 (6 punti)

- Scrivere un metodo iterativo  
**public static boolean equilibrata (Elem lista)**  
che, presa come parametro una lista di numeri interi, restituisce **true** se e solo se **lista** è equilibrata.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **60** minuti per ogni parte.
- **Chi svolge la seconda prova parziale deve consegnare dopo 60 minuti.**
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco il proprio libretto o altro documento di identità.

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

# Programmazione 1

## A.A. 2013/2014

### Appello del 17 Dicembre 2013 – Compito n° 4

#### Prima parte

#### Esercizio 1 (12 punti)

1.1 Cosa stampa il seguente frammento di codice Java?

```
int [] A = {3, -78, -90, 200, -15, -17, 8};
int j, i = 4;
for (j=0; j<i; j++) {
    if (A[j]>0) i=i+1;
}
System.out.println(A[j]);
```

1.2 Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(2));
    System.out.println(enigma(1));
    System.out.println(enigma(2511));
    System.out.println(enigma(1050));
}
public static int enigma (int x){
    if (x==0) return 0;
    return 6 - enigma (x-1);
}
}
```

#### Esercizio 2 (14 punti)

Un array si dice **equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre equilibrati.

Ad esempio, l'array [1, 1, 10, 4, 6, 2] è equilibrato in quanto  $1+1+10 = 4+6+2$ .

Scrivere un metodo iterativo

**public static boolean equilibrato (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è equilibrato.

#### Esercizio 3 (8 punti)

Un array si dice **totalmente equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà, ed inoltre anche il sottoarray costituito dalla sua prima metà e quello costituito dalla sua seconda metà sono totalmente equilibrati (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre totalmente equilibrati.

Ad esempio, l'array [1, 3, 1, 2, 1, 2] è totalmente equilibrato in quanto  $1+3+1 = 2+1+2$ , ed inoltre anche i sottoarray [1, 3, 1] e [2, 1, 2] sono totalmente equilibrati poiché  $1=1$  e  $2=2$ , rispettivamente; infine, i sottoarray [1], [1], [2] e [2] sono banalmente totalmente equilibrati poiché composti da un solo elemento.

Scrivere un metodo ricorsivo

**public static boolean totalmenteEquilibrato (int[] a)**

che, sfruttando il metodo dell'esercizio precedente, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è totalmente equilibrato.

## Seconda parte

### Esercizio 4 (16 punti)

Si consideri il seguente tipo di dati:

```
class Riga {
    int lunghezza;
    int somma;
}
```

Scrivere un metodo

**public static Riga[] analizza(int[][] a)**

che preso come parametro un array bidimensionale **a** con  $n$  righe di numeri interi (in cui ogni riga non è necessariamente della stessa lunghezza), restituisce un array di Riga di lunghezza uguale a  $n$ , e tale che l'elemento in posizione  $i$  ( $i=0, \dots, n-1$ ) contiene

- nel campo `lunghezza` la lunghezza dell'array (monodimensionale) che costituisce la riga  $i$  di **a**, e
- nel campo `somma` la somma degli elementi di tale riga.

Ad esempio, se **a** = **{{-10, 35}, {-20}, {-14, 2, 19}}** il metodo deve restituire un array di lunghezza 3 in cui

- in posizione 0 c'è un'istanza della classe `Riga` con `lunghezza=2` e `somma=25`,
- in posizione 1 c'è un'istanza della classe `Riga` con `lunghezza=1` e `somma=-20`,
- in posizione 2 c'è un'istanza della classe `Riga` con `lunghezza=3` e `somma=7`.

### Esercizio 5 [non è possibile utilizzare nessun metodo scritto a lezione senza riscriverlo completamente]

Si consideri il seguente tipo di dati visto a lezione, che rappresenta una lista di numeri interi, identificata dal suo primo elemento.

```
class Elem {
    int valore;
    Elem next;
}
```

#### 5.1 (6 punti)

- Scrivere un metodo iterativo

**public static int massimaDifferenza (Elem lista)**

che presa come parametro una lista di interi, restituisce la differenza tra il massimo ed il minimo valore in essa contenuti. Se la lista è vuota, viene restituito 0.

#### 5.2 (6 punti)

- Scrivere un metodo ricorsivo

**public static int lunghezza (Elem lista)**

che presa come parametro una lista di interi, restituisce la sua lunghezza.

Una lista si dice **sbilanciata a destra** se la somma degli elementi nella sua prima metà è minore della somma degli elementi nella sua seconda metà (qualora la lista abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Una lista vuota o una lista con un solo elemento sono sempre sbilanciate a destra.

Ad esempio, la lista  $1 \rightarrow 3 \rightarrow 4 \rightarrow 4 \rightarrow 6 \rightarrow 2$  è sbilanciata a destra in quanto  $1+3+4 < 4+6+2$ .

#### 5.3 (6 punti)

- Scrivere un metodo iterativo

**public static boolean sbilanciataDx (Elem lista)**

che, presa come parametro una lista di numeri interi, restituisce **true** se e solo se **lista** è sbilanciata a destra.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **60** minuti per ogni parte.
- **Chi svolge la seconda prova parziale deve consegnare dopo 60 minuti.**
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato e depositato in vista sulla cattedra**
- Mettere in vista sul banco il proprio libretto o altro documento di identità.