

Cognome e Nome _____

Matricola _____

Programmazione 1

A.A. 2015/2016

Appello del 2 Febbraio 2016

Compito n° 1

Esercizio 1 (8 punti)

1.1 (4 punti) Cosa stampa il seguente frammento di codice Java?

```
int [] A = {10,2,30,50};
for (int i=A.length-1;i>=0;i--) {
    for (int j=i;j<A.length;j++){
        System.out.print((i+A[j]) + " ");
    }
    System.out.println();
}
```

1.2 (4 punti) Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(1));
    System.out.println(enigma(3));
    System.out.println(enigma(6));
}

static int enigma (int x){
    if (x<=0) return 0;
    return 2*x + enigma (x-2);
}
}
```

Esercizio 2 (8 punti)

static int[] estraiIntervallo(int[] a, int min, int max)

che, preso come parametro un array **a** di numeri interi, e due interi **min** e **max**, restituisce un array di numeri interi (della opportuna lunghezza) contenente tutti e soli gli elementi di **a** che sono compresi tra **min** e **max** (estremi inclusi), rispettando l'ordine e il numero di occorrenze di **a**.

Se nessun elemento di **a** rispetta tali condizioni, viene restituito un array di lunghezza 0.

Se **a** vale *null*, viene restituito *null*.

Ad esempio, se **a**={1, 4, 3, -7, 5, 6, 3, 8, 4, -4}, **min** vale **-3** e **max** **5** viene restituito l'array {1, 4, 3, 5, 3, 4}.

Esercizio 3 (8 punti)

Si consideri il tipo di dato

```
class Durata {
    int hh; //le ore
    int mm; //i minuti
    int ss; //i secondi
}
```

che rappresenta una durata (i minuti ed i secondi variano tra 0 a 59).

- **(4 punti)** Scrivere un metodo **static Durata somma(Durata d1, Durata d2)** che, prese come parametri due durate **d1** e **d2**, crea e restituisce una nuova durata pari alla somma delle durate **d1** e **d2**. Se **d1** e/o **d2** valgono *null*, viene restituito *null*.
Ad esempio, se **d1** vale 2 ore, 50 minuti e 40 secondi e **d2** vale 0 ore, 15 minuti e 30 secondi, la somma deve valere 3 ore, 6 minuti e 10 secondi.
- **(4 punti)** Scrivere un metodo iterativo **static int contaBrevi(Durata[] a)** che, preso come parametro un array di Durata **a**, restituisce il numero di elementi di **a** che durino al più un'ora. Se **a** vale *null*, viene restituito 0. Il metodo deve inoltre gestire in modo opportuno gli elementi dell'array che valgono *null* (non facendoli rientrare nel conteggio).

Esercizio 4 (8 punti)

Si consideri il seguente tipo di dato, che rappresenta una **pila** di numeri interi:

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

```
class Pila{
    Elem top;
    int size;
}
```

- **(4 punti)** Scrivere un metodo iterativo **static void multiPush(Pila p, int[] a)** che, presa come parametro una pila **p** di numeri interi e un array di interi **a**, inserisce nella pila tutti gli elementi presenti in **a** in ordine inverso rispetto alla posizione in **a** (in modo che l'elemento in posizione **0** in **a** si trovi in cima alla pila).
Se la pila **p** non esiste, il metodo non fa nulla.
Se l'array **a** non esiste o è vuoto, la pila non viene modificata.
- **(4 punti)** Scrivere un metodo iterativo **static int[] multiPop(Pila p, int n)** che, presa come parametro una pila **p** di numeri interi e un intero **n**, estrae dalla pila **n** elementi (se $n > p.size$ vengono estratti $p.size$ elementi) e crea e restituisce un array di lunghezza uguale al numero di elementi estratti, contenente in posizione **0** il primo elemento estratto, in posizione **1** il secondo estratto e così via (in posizione **i** contiene l'(i+1)-esimo elemento estratto).
Se la pila non esiste oppure è vuota, viene restituito un array di lunghezza **0**.

Si possono sfruttare (richiamandoli senza scriverli) tutti i metodi visti a lezione.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **90** 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. 2015/2016

Appello del 2 Febbraio 2016

Compito n° 2

Esercizio 1 (8 punti)

1.1 (4 punti) Cosa stampa il seguente frammento di codice Java?

```
int [] A = {20,10,35,40};
for (int i=A.length-1;i>=0;i--) {
    for (int j=i;j<A.length;j++){
        System.out.print((i+A[j]) + " ");
    }
    System.out.println();
}
```

1.2 (4 punti) Cosa stampa il seguente programma Java?

```
public class Main {
public static void main(String[] args) {
    System.out.println(enigma(1));
    System.out.println(enigma(4));
    System.out.println(enigma(6));
}

    static int enigma (int x){
        if (x<=0) return 0;
        return 3*x + enigma (x-3);
    }
}
```

Esercizio 2 (8 punti)

static int[] estraiIntervallo(int[] a, int min, int max)

che, preso come parametro un array **a** di numeri interi, e due interi **min** e **max**, restituisce un array di numeri interi (della opportuna lunghezza) contenente tutti e soli gli elementi di **a** che sono compresi tra **min** e **max** (estremi inclusi), rispettando l'ordine e il numero di occorrenze di **a**.

Se nessun elemento di **a** rispetta tali condizioni, viene restituito un array di lunghezza 0.

Se **a** vale *null*, viene restituito *null*.

Ad esempio, se **a**={1, 4, 3, -7, 5, 6, 3, 8, 4, -4}, **min** vale **-3** e **max** **5** viene restituito l'array {1, 4, 3, 5, 3, 4}.

Esercizio 3 (8 punti)

Si consideri il tipo di dato

```
class Durata {
    int hh; //le ore
    int mm; //i minuti
    int ss; //i secondi
}
```

che rappresenta una durata (i minuti ed i secondi variano tra 0 a 59).

- **(4 punti)** Scrivere un metodo **static Durata somma(Durata d1, Durata d2)** che, prese come parametri due durate **d1** e **d2**, crea e restituisce una nuova durata pari alla somma delle durate **d1** e **d2**. Se **d1** e/o **d2** valgono *null*, viene restituito *null*.
Ad esempio, se **d1** vale 2 ore, 50 minuti e 40 secondi e **d2** vale 0 ore, 15 minuti e 30 secondi, la somma deve valere 3 ore, 6 minuti e 10 secondi.
- **(4 punti)** Scrivere un metodo iterativo **static int contaBrevi(Durata[] a)** che, preso come parametro un array di Durata **a**, restituisce il numero di elementi di **a** che durino al più un'ora. Se **a** vale *null*, viene restituito 0. Il metodo deve inoltre gestire in modo opportuno gli elementi dell'array che valgono *null* (non facendoli rientrare nel conteggio).

Esercizio 4 (8 punti)

Si consideri il seguente tipo di dato, che rappresenta una **pila** di numeri interi:

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

```
class Pila{
    Elem top;
    int size;
}
```

- **(4 punti)** Scrivere un metodo iterativo **static void multiPush(Pila p, int[] a)** che, presa come parametro una pila **p** di numeri interi e un array di interi **a**, inserisce nella pila tutti gli elementi presenti in **a** in ordine inverso rispetto alla posizione in **a** (in modo che l'elemento in posizione **0** in **a** si trovi in cima alla pila).
Se la pila **p** non esiste, il metodo non fa nulla.
Se l'array **a** non esiste o è vuoto, la pila non viene modificata.
- **(4 punti)** Scrivere un metodo iterativo **static int[] multiPop(Pila p, int n)** che, presa come parametro una pila **p** di numeri interi e un intero **n**, estrae dalla pila **n** elementi (se $n > p.size$ vengono estratti $p.size$ elementi) e crea e restituisce un array di lunghezza uguale al numero di elementi estratti, contenente in posizione **0** il primo elemento estratto, in posizione **1** il secondo estratto e così via (in posizione **i** contiene l'(i+1)-esimo elemento estratto).
Se la pila non esiste oppure è vuota, viene restituito un array di lunghezza **0**.

Si possono sfruttare (richiamandoli senza scriverli) tutti i metodi visti a lezione.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **90** 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.