Programmazione e Algoritmi 1

A.A. 2023/24 — Compito dell'11 settembre 2024 – Versione Java

prof. Gianluca Amato

Gli esercizi di programmazione saranno valutati sulla base della correttezza, efficienza e comprensibilità della soluzione proposta. In generale, **se volete usare una funzione o un metodo che non è stato presentato a lezione, chiedete prima al docente se è consentito**.

Esercizio 1 (5 punti)

Descrivere gli algoritmi di ricerca lineare e di ricerca binaria. Per ognuno di essi:

- specificare cosa si aspetta in input l'algoritmo;
- specificare cosa produce come risultato l'algoritmo;
- illustrarne a grandi linee il funzionamento;
- indicarne la complessità computazionale.

Determinare inoltre due input specifici, validi per entrambi gli algoritmi: uno in cui è più efficiente (fa meno confronti) la ricerca lineare, ed uno in cui è più efficiente la ricerca binaria (e spiegare brevemente il motivo).

Esercizio 2 (8 punti)

Si consideri il seguente programma Java, e se ne scriva la traccia di esecuzione negli appositi moduli.

```java
public class Esercizio2 {

    public static boolean f1(int n) {
        if (n == 0)
            return true;
        else
            return f2(n - 1);
    }

    public static boolean f2(int n) {
        if (n == 0)
            return false;
        else
            return f1(n - 1);
    }

    public static void main(String[] args) {
        System.out.println(f1(4));
    }
}
```

Esercizio 3 (5 punti)

Scrivere una metodo static `squares` che prende come parametro un numero intero n (che possiamo supporre essere maggiore o uguale a 0) e restituisce l'array di tutti i quadrati perfetti minori o uguali ad n. Ad esempio, `squares(30)` restituisce `{0, 1, 4, 9, 16, 25}`. Si ricorda che un quadrato perfetto è un numero del tipo x^2 per qualche intero x, ed infatti 0, 1, 4, 9, 16 e 25 sono rispettivamente 0^2, 1^2, 2^2, 3^2, 4^2 e 5^2.

Esercizio 4 (5 punti)

Scrivere due test nel framework `JUnit` per verificare il corretto funzionamento di `squares`. In particolare, si implementino due test:

- un test che verifichi se le seguenti chiamate di funzione restituiscono il risultato desiderato:
  - `squares(5)`
  - `squares(1)`
- un test per il caso in cui il numero n è un intero positivo o nullo generato casualmente. In questo caso, il test deve controllare solo che l'array restituito dalla funzione non sia vuota e che l'ultimo elemento sia minore o uguale ad n.

Esercizio 5 (5 punti)

Si utilizzi il metodo static `squares` per scrivere un programma che legge un numero intero n da tastiera e stampa la radice quadrata di n, approsimata all'intero più piccolo (ad esempio, se l'input immesso da tastiera è 30 il risultato sarà 5). In aggiunta, se il numero immesso è negativo, il programma deve stampare un messaggio di errore e chiedere di reinserire il numero. Questo è un esempio di interazione: in rosso quello scritto dall'utente, in nero l'output del programma.

```
Immetti un numero positivo o nullo: -5
Errore: il numero deve essere positivo o nullo.
Immetti un numero positivo o nullo: 30
La radice quadrata intera di 30 è 5.
```

Esercizio 6 (5 punti)

Si consideri la seguente classe Java:

```
class Esame {
    String nome;
    int voto;
}
```

Scrivere un metodo statico `classificaEsami` che prende come parametro un array di oggetti di classe `Esame` e restituisce un array di stringhe `a` tale che, alla posizione i-esima di `a`, si trova il nome di un esame che ha esattamente quel voto, o il valore `null` se questo esame non esiste. In caso di presenza di più esami con uno stesso voto, si restituisca il nome del primo esame trovato.

Ad esempio, se l'array di esami è `{new Esame("Economia", 3), new Esame("Informatica", 1), new Esame("Matematica", 3), new Esame("Diritto", 0}`, allora un possibile risultato del metodo `classificaEsami` è l'array `{"Diritto", "Informatica", null, "Economia"}`.