

# Corso di Laurea Ingegneria Informatica

## Fondamenti di Informatica 1

---

**Dispensa E04**

**Esempi di algoritmi e programmi**

---

**C. Limongelli - A. Miola**

Settembre 2006

# Contenuti

---

- ❑ **Somma di una sequenza di numeri interi**
- ❑ **Lunghezza di una sequenza**
- ❑ **Massimo di una sequenza di dieci numeri (\*)**
- ❑ **Somma dei pari e dei dispari in una sequenza**
- ❑ **Calcolo degli zeri di una sequenza di numeri interi (\*)**
- ❑ **Che giorno sarà ? (\*)**
- ❑ **Somma dei primi N numeri interi positivi**
- ❑ **La formula di Gauss**
- ❑ **Generazione di N numeri interi random (\*)**
- ❑ **Escursione dei valori di una sequenza di numeri interi (\*)**
- ❑ **Calcolo dei divisori di un numero intero (\*)**
- ❑ **Verifica se un numero è primo (\*)**

# Nota Bene

---

- **Gli esercizi in ROSSO (cioè senza asterisco) sono sul libro di testo a cui si rinvia**

# Esercizi

- ❑ **Verificare l'algoritmo per il calcolo della lunghezza di una sequenza**
  - **ad esempio, se la sequenza è 1 3 5 0**
- ❑ **Implementare l'algoritmo per il calcolo della lunghezza di una sequenza**
- ❑ **Scrivere un algoritmo che legge una sequenza non vuota di numeri interi e ne calcola la media aritmetica**
  - **la somma degli elementi diviso il numero degli elementi**

# Calcolo del massimo di dieci numeri . . .

## □ Si consideri il seguente problema di ingresso-uscita

- si vuole leggere dalla tastiera una sequenza di numeri interi di lunghezza fissata (ad esempio, di lunghezza 10), calcolare il valore massimo degli elementi e visualizzarlo sullo schermo
- **Scrivi una sequenza di dieci numeri interi**  
**1 12 -13 2 0**  
**-4 8 0 4 10**
- **Il valore massimo calcolato è : 12**

# ... Calcolo del massimo di dieci numeri

## □ Problema

- lettura di una sequenza di 10 numeri interi e calcolo del valore massimo degli elementi

## □ Dati di ingresso

- una sequenza di dieci numeri interi  $A_1, A_2, \dots, A_{10}$

## □ Pre-condizione

- nessuna

## □ Dati di uscita

- un numero max

## □ Post-condizione

- max è il massimo degli elementi della sequenza  $A_1, A_2, \dots, A_{10}$

# Algoritmo per il calcolo del massimo tra dieci numeri interi . . .

- L'algoritmo può essere realizzato gestendo le seguenti variabili
  - **num** - valore dell'elemento corrente della sequenza
  - **cont** - contatore degli elementi già letti
  - **max** - valore massimo degli elementi già letti
  
- La gestione delle variabili **max** e **cont** avviene come segue
  - inizialmente **cont** vale **0**
  - leggi il primo numero **num**, assumilo come valore provvisorio del massimo **max**, e incrementa **cont**
  - per ogni altro numero **num** letto
    - **max** deve essere confrontato con **num** ed eventualmente modificato
    - **cont** deve essere incrementata di uno

# ... Algoritmo per il calcolo del massimo tra dieci numeri interi

Leggi una sequenza di dieci numeri interi e calcolane il massimo:

1. *inizializza il contatore dei numeri letti*

2. *leggi un numero **num** da input*

3. *inizializza **max***

4. *incrementa il contatore dei numeri letti*

5. *finché ci sono ancora numeri da leggere*

5.1 *leggi un numero **num***

5.2 *confronta **num** con **max***

*se **num** > **max***

*memorizza il contenuto*

*di **num** in **max***

5.3 *incrementa **cont***

6. *il risultato è **max***

```
cont = 0;
```

```
...
```

```
max = num;
```

```
cont = cont + 1;
```

```
while (cont <= 9)
```

```
...
```

```
if (num > max)
```

```
max = num;
```

```
cont = cont + 1;
```



# Applicazione per il calcolo del massimo tra dieci numeri interi . . .

```
import fiji.io.*;
/* Applicazione che calcola il massimo tra 10 numeri */

class Max10Numeri{
    public static void main(String[] args){
        int num;           //numero letto
        int max;           //massimo calcolato
        int cont;          //contatore dei numeri letti

        //precondizioni sequenza non vuota: almeno un elemento

        /* inizializzazione del contatore */
        cont = 0;

        /* lettura del primo numero - fuori dal ciclo*/
        System.out.println("Scrivi una sequenza di 10 numeri interi");
        num = Lettore.in.leggiInt();
        /* appena leggo un numero incremento il contatore */
        cont = cont + 1;

        . . .
    }
}
```

# ... Applicazione per il calcolo del massimo tra dieci numeri interi

. . .

```
/* essendo l'unico numero letto esso è il massimo */  
max = num;
```

```
/* devono essere letti altri 9 numeri */
```

```
while (cont <= 9){  
    /*lettura di un numero ad ogni iterazione */  
    num = Lettore.in.leggiInt();  
    cont = cont + 1;  
    /* se num è maggiore del max, aggiornò max */  
    if (num > max)  
        max = num;
```

```
}
```

```
System.out.print("valore massimo calcolato è: ");
```

```
System.out.println(max);
```

```
}
```

```
}
```

# Calcolo degli zeri di una sequenza

## □ Risolvere il seguente problema di ingresso-uscita

- leggere dalla tastiera una sequenza di numeri interi, disposti su una singola linea e separati da spazi, calcolare il numero di zeri della sequenza (ovvero, il numero di elementi della sequenza che sono uguali a zero) e visualizzare il numero di zeri sullo schermo

Scrivi una sequenza di numeri interi

**10 15 0 -2**

Il numero di zeri nella sequenza è 1

***... bisogna usare l'istruzione if ...***

# Calcolo degli zeri di una sequenza – frammento di codice da completare

```
int numero;    // elemento corrente della sequenza
int zeri;     // numero di zeri nella sequenza
/* leggi una sequenza di numeri interi e
 * calcolane il numero di zeri */
/* inizialmente zeri vale zero */
zeri = 0;
/* finché ci sono altri elementi nella sequenza,
 * leggili e calcola il numero di zeri */
while (!Lettore.in.eoln()) {
    /* leggi un elemento della sequenza */
    numero = Lettore.in.leggiInt();
    . . . // da completare
}
/* visualizza zeri */
System.out.println(zeri);
```

# Che giorno sarà? . . .

---

## □ Problema

- Applicazione che calcola il giorno della settimana corrispondente ad una data

## □ Esempio

- sapendo che il primo gennaio del 2007 è lunedì, che giorno sarà il 15 agosto? (mercoledì)

# . . . Che giorno sarà?

## □ **Dati di ingresso**

- Il giorno della settimana corrispondente al 1° gennaio dell'anno di riferimento;
- L'anno di riferimento;
- La data (gg/mm) appartenente all'anno di riferimento di cui si vuole conoscere il giorno della settimana;

## □ **Pre-condizione**

- L'anno è un intero positivo maggiore di 1600
- La data deve essere ammissibile.

## □ **Dati di uscita**

- Il giorno della settimana corrispondente alla data immessa in input

# Convenzione

□ I giorni della settimana sono rappresentati da caratteri (tipo **char**) con la seguente convenzione:

<b>Giorni</b>	<b>char</b>
Lunedì	<b>l</b>
Martedì	<b>m</b>
Mercoledì	<b>M</b>
Giovedì	<b>g</b>
Venerdì	<b>v</b>
Sabato	<b>s</b>
Domenica	<b>d</b>

# L'algoritmo

- ❑ Leggo i dati da input.
- ❑ Calcolo il numero di giorni (**numGiorni**) che separano la data di interesse dal 1° gennaio.
- ❑ Il resto della divisione di **numGiorni** per 7 indica la distanza settimanale del giorno dalla data di interesse
- ❑ Copio il giorno del 1° gennaio in **giornoRes** che rappresenterà il risultato finale
- ❑ Finché (**cont** <= distanza settimanale)
  - Calcolo il giorno successivo a **giornoRes**
  - Assegno il valore del nuovo giorno a **giornoRes**
- ❑ Stampa dei risultati



# Il numero dei giorni . . .

---

□ Per calcolare il numero di giorni che separano il primo giorno dell'anno e l'ultimo giorno del mese precedente alla data di interesse, sfruttiamo il metodo **NumeroGiorni**, che tra le altre cose terrà conto se un anno è bisestile o meno

# ... Il numero dei giorni

```
public static int NumeroGiorni(int mese, int anno){
    int i;
    int numGiorni;
    i=1;
    numGiorni=0;
    while (i< mese) {
        /* calcolo dei giorni del mese */
        if (i==2)
            if (Bisestile.bisestile(anno))
                numGiorni = numGiorni+29;
            else numGiorni = numGiorni+28;
        else //mesi con 30 giorni
            if (i==4 || i==6 || i==9 || i==11)
                numGiorni = numGiorni+30;
            else numGiorni = numGiorni+31;
        i=i+1;
    }// end while
    return numGiorni;
}
```

# Il giorno successivo . . .

---

□ L'individuazione del giorno successivo viene effettuata tramite il metodo **succ** in funzione del parametro in input di tipo **char**, che rappresenta un giorno, questo metodo restituisce la codifica del giorno successivo.

# . . . Il giorno successivo

```
public static char succ(char giorno){
    char successore;
    if (giorno == 'l')
        successore = 'm';
    else if(giorno == 'm')
        successore = 'M';
    else if (giorno == 'M')
        successore = 'g';
    else if (giorno == 'g')
        successore = 'v';
    else if (giorno == 'v')
        successore = 's';
    else if (giorno == 's')
        successore = 'd';
    else if (giorno == 'd')
        successore = 'l';
    else successore = 'e'; // per gestire un errore
    return successore;
}
```

# Stampa del Giorno

---

□ Il metodo **StampaGiorno** viene impiegato per passare dalla rappresentazione dei giorni tramite **char** ad una rappresentazione più leggibile

# ...Stampa del Giorno

```
public static void StampaGiorno(char c){
    if (c == 'l')
        System.out.println(" lunedì");
    else if (c == 'm')
        System.out.println(" martedì");
    else if (c == 'M')
        System.out.println(" mercoledì");
    else if (c == 'g')
        System.out.println(" giovedì");
    else if (c == 'v')
        System.out.println(" venerdì");
    else if (c == 's')
        System.out.println(" sabato");
    else if (c == 'd')
        System.out.println(" domenica");
    else
        System.out.println(" errore!! ");
}
```

# La classe CheGiornoSara: dichiarazioni

```
import fiji.io.*;
class CheGiornoSara{
    public static void main(String[] args){

        /*input*/
        char giorno1gen;           //giorno della settimana
                                   //corrispondente al 1 gennaio
        int anno;                 // anno corrispondente
        int giorno, mese;        // giorno e mese di cui si vuol
                                   // conoscere il giorno della
                                   // settimana

        /*output*/
        char giornoRis;           // giorno della settimana
                                   //corrispondente a g,m

        /* variabili d'appoggio */
        int numGiorni; //numero di giorni tra il 1 gennaio e la
                                   //data richiesta

        int i,app;
```

# La classe CheGiornoSara: acquisizione dei dati

```
/*lettura della data di partenza senza controllo sulla  
correttezza dei dati */
```

```
    System.out.println("Scrivi il giorno della settimana  
corrispondente al 1 gennaio");  
    System.out.println("(l m M g v s d) ");  
    giorno1gen = Lettore.in.leggiChar();
```

```
    System.out.println("Di quale anno?");  
    anno = Lettore.in.leggiInt();
```

```
    System.out.println("Di quale data vuoi calcolare il  
giorno della settimana? ");
```

```
    System.out.println("scrivi giorno e mese in cifre,  
separati da spazi");
```

```
    giorno = Lettore.in.leggiInt();  
    mese = Lettore.in.leggiInt();
```



# La classe CheGiornoSara: calcolo

```
/* calcolo il numero di giorni tra il 1 gennaio e l'ultimo
giorno del mese precedente */
    numGiorni=NumeroGiorni(mese,anno);

/* poi aggiungo i giorni del mese in esame */
numGiorni = numGiorni+giorno-1;
/* -1 perchè devo contare dal 1 gennaio escluso */

app = numGiorni % 7; // distanza settimanale da colmare
giornoRis= giorno1gen;
i=1;
while(i<=app){
    giornoRis=succ(giornoRis);
    i++
}
```

# La classe CheGiornoSara: stampa del risultato

---

```
    /* stampa dei dati */  
    System.out.print(" il giorno corrispondente al "  
+ giorno + "/" + mese + "/" + anno + " è: ");  
    StampaGiorno(giornoRis);  
  
} //end main
```

# Generazione di n numeri interi random

## □ Problema

- Scrivere un'applicazione per generare e visualizzare sullo schermo N numeri interi random compresi tra 0 e 99

## □ Dati di ingresso

- un intero N

## □ Pre-condizione

- $N \geq 0$

## □ Dati di uscita

- Una sequenza di numeri

## □ Post-condizione

- Nessuna

# Algoritmo per la generazione di n interi numeri random

- ❑ Il metodo **Math.random()** calcola un numero pseudo-casuale di tipo **double** compreso nell'intervallo [0,1).
- ❑ Bisogna trasformarlo in un numero di tipo **int** compreso in [0,100) **moltiplicandolo per 100**
- ❑ Leggi un intero **n**
- ❑ Inizializza un **contatore i** a **0**, che permette di contare il numero di volte che deve essere richiamato il metodo **Math.random()** per generare gli **n** numeri random
- ❑ **Mentre  $i < n$** 
  - Genera un numero random, moltiplicalo per 100, trasformalo in un intero e visualizzalo sullo schermo
  - Incrementa il contatore **i**

# Codifica per la generazione di n interi numeri random

```
//pre: n>=0
int res; // numero generato
int n; // numero di elementi da generare
int i; // contatore per l'istr. ripetitiva

/*lettura del valore di n */
System.out.println("quanti numeri vuoi generare? ");
n = Lettore.in.leggiInt();

i=0;
while(i <n ){
    /* calcolo del numero random */
    res = (int)(Math.random()*100);
    System.out.print(res);
    System.out.print(" ");
    i=i+1;
}
```

# Applicazione per la generazione di n interi numeri random

```
import fiji.io.*;
/* Applicazione che genera n numeri random tra 0 e 100 */
class NumeriRandom{
    public static void main(String[] args){
        //pre: n>=0
        int res; // numero generato
        int n; // numero di elementi da generare
        int i; // contatore per l'istr. ripetitiva
        /*lettura del valore di N */
        System.out.println("quanti numeri vuoi generare? ");
        n = Lettore.in.leggiInt();
        i=0;
        while( i<n ){
            /* calcolo del numero random */
            res = (int)(Math.random()*100);
            System.out.print(res);
            System.out.print(" ");
            i=i+1;
        }
        System.out.println();
    }
}
```

# Esercizi

- ❑ **Definire un metodo di un oggetto** per generare e visualizzare N numeri interi random nell'intervallo [0,100), assumendo il valore di N come parametro del metodo
- ❑ **Definire un metodo di un oggetto** per calcolare e visualizzare il valore massimo di una sequenza di N numeri interi, assumendo il valore di N come parametro del metodo
- ❑ **Definire una classe applicazione** per generare N numeri interi random nell'intervallo [0,100) e per calcolare e visualizzare il valore massimo degli N numeri interi generati in modo random

# Escursione dei valori in una sequenza

## ❑ Problema

- Scrivere un'applicazione che calcola l'escursione dei valori di una sequenza di interi letta da tastiera

## ❑ Esempio

- Dati i valori 3 -1 2 4 -2 l'escursione è 7 (da -2 a 4)

## ❑ Dati di ingresso

- una sequenza di numeri interi

## ❑ Pre-condizione

- La sequenza è non vuota

## ❑ Dati di uscita

- Massimo, minimo ed escursione

## ❑ Post-condizione



# Algoritmo per l'escursione dei valori in una sequenza

- Inizializzo **max** e **min**:
  - Leggo il primo elemento della sequenza e lo memorizzo in **num**
  - **max = num**
  - **min = num**
- Finché non sono terminati gli elementi sulla linea di input
  - leggo un numero e lo memorizzo in **num**
  - se **num > max**
    - allora aggiorno **max** con **num**
  - se **num < min**
    - allora aggiorno **min** con **num**
- L'escursione è data da (**max-min+1**)

*Se la sequenza è composta tutta da valori negativi ?*

# Codifica per l'escursione dei valori in una sequenza

```
System.out.println("Scrivi una sequenza non vuota di
                    numeri interi ");

num = Lettore.in.leggiInt();
max = num;
min = num;
/* finchè non viene dato invio */
while (!Lettore.in.eoln()){
    /*lettura di un numero ad ogni iterazione */
    num = Lettore.in.leggiInt();
    /* se num è maggiore del max, aggiorno max */
    if (num > max)
        max = num;
    /* se num è minore di min aggiorno min */
    if (num < min)
        min = num;
}
System.out.print("L'escursione della sequenza è ");
System.out.println(Math.abs(max-min+1));
```

# Applicazione per l'escursione dei valori in una sequenza . . .

```
import fiji.io.*;
/* Applicazione che calcola l'escursione dei valori di una
sequenza di interi letta da input */
class Escursione{

    public static void main(String[] args){
        int num;
        int max,min;
        //precondizioni sequenza non vuota
        System.out.println("Scrivi una sequenza non vuota di
                            numeri interi");

        num = Lettore.in.leggiInt();
        max=num;
        min=num;
        /* OPPURE */
        /* inizializzazione delle var con il massimo e minimo
        intero rappresentabili: valori che saranno subito
        aggiornati dalla lettura del primo elemento che
        esiste perchè la sequenza è non vuota */
        //max =Integer.MIN_VALUE;
        //min =Integer.MAX_VALUE;
        /** sono inizializzazioni sufficientemente generali? **/
    }
}
```

# ... Applicazione per l'escursione dei valori in una sequenza

```
/* finchè non viene dato invio */
while (!Lettore.in.eoln()){
    /* lettura di un numero ad ogni iterazione */
    num = Lettore.in.leggiInt();
    /* se num è maggiore del max, aggiornò max */
    if (num > max)
        max = num;
    /* se num è minore di min aggiornò min */
    if (num < min)
        min = num;
}
System.out.print("valore massimo ");
System.out.println(max);
System.out.print("valore minimo  ");
System.out.println(min);
System.out.print("L'escursione della sequenza è ");
System.out.println(Math.abs(max-min+1));
}
}
```

# Calcolo dei divisori di un numero intero

## □ Problema

- Dato un numero **num** intero positivo calcolare e visualizzare tutti i suoi divisori (**1** e **num** esclusi)

## □ Esempio

- I divisori di 28 sono 2, 4, 7, e 14

## □ Dati di ingresso

- **num**

## □ Pre-condizione

- **num** intero positivo

## □ Dati di uscita

- **sequenza dei divisori**

## □ Post-condizione

- **???**

# Algoritmo per il calcolo dei divisori di un numero intero

Dato un numero intero **num**, verifico che tutti i valori **div** compresi tra 2 e **num/2** dividano o meno **num**

Se il risultato della divisione di **num** per **div** dà resto 0 allora **div** è un divisore di **num** e lo visualizzo

- inizializza **lim** a **num/2**
- Inizializza **div** a 2
- Finché il divisore **div** è  $< \text{lim}$ 
  - Se il resto della divisione di **num** per **div** è 0  
allora visualizza **div**
  - incrementa **div** di 1

# Codifica per il calcolo dei divisori di un numero intero

```
/* lettura del numero */
System.out.println("Scrivi un intero positivo");
num = Lettore.in.leggiInt();
/* inizializzo il limite massimo dei divisori */
lim = num/2;
/* inizializzo il divisore */
div = 2;

/* spiegazione delle stampe che verranno prodotte
   all'interno del ciclo while */
System.out.println("I suoi divisori sono: ");

/* verifico da 2 fino a div */
while (div <= lim){
    if (num%div == 0)
        System.out.println(div);
    div = div + 1;
}
```

# Applicazione per il calcolo dei divisori di un numero intero . . .

```
import fiji.io.*;
/* Applicazione che calcola e visualizza i divisori di un
 * numero intero letto da tastiera */
class Divisori{

    public static void main(String[] args){
        int num;           //numero letto
        int lim;           //limite massimo dei divisori
        int div;           //divisore

        //precondizioni: num è intero positivo (>=1)
        /* dato num, verifico che tutti i valori div compresi
         * tra 2 e num/2 dividano o meno num. Se il risultato
         * della divisione di num per div dà resto 0 allora
         * div è un divisore di num e lo visualizzo */

        /*lettura del numero*/
        System.out.println("Scrivi un intero positivo");
        num = Lettore.in.leggiInt();
    }
}
```



# ... Applicazione per il calcolo dei divisori di un numero intero

```
/* inizializzo il limite massimo dei divisori */
lim = num/2;

/* inizializzo il divisore */
div = 2;

/* spiegazione delle visualizzazioni che verranno
 * prodotte all'interno del ciclo while */
System.out.println("I suoi divisori sono: ");

/* verifico i divisori da 2 fino a lim */
while (div <= lim){
    if (num%div == 0)
        System.out.println(div);
    div = div + 1;
}
}
```

# Esercizi

---

- Scrivere un'applicazione che verifica se un numero intero letto da tastiera è primo**
  
- Scrivere un metodo che verifica se un numero intero assunto come parametro del metodo è primo**
  
- Scrivere un'applicazione che calcola e visualizza i primi 10 numeri primi**

# Esercizio: verifica se un numero è primo

## □ Problema

- Scrivere un'applicazione che verifica se un numero è primo

## □ Dati di ingresso

- Un intero positivo  $n$

## □ Pre-condizione

- $n > 0$

## □ Dati di uscita

- Vero/falso (primo / non primo)

## □ Post-condizione

# Algoritmo per la verifica se un numero è primo

---

Leggo **n** da input

Inizializzo **div** a 2

Inizio a provare se **div** divide **n**:

se **div** divide **n** mi fermo perché **n** non è primo

Finché (**div** non divide **n**) e (**div** non è arrivato a **n/2**)

incremento **div** di 1

se esco dal ciclo perché **n** è divisibile per **div**

allora **n** è primo

altrimenti **n** non è primo

# Codifica per la verifica se un numero è primo

```
int div; //divisore per la verifica
div = 2;

while ((n%div!=0)&& (div<=n/2))
    div = div +1;
/* esco dal ciclo quando n%div = 0 (il num è
   divisibile) oppure quando div > n/2
   (li ho provati tutti) */

if (n%div==0)
    return false;
else
    return true;
```

# Applicazione per la verifica se un numero è primo . . .

```
import fiji.io.*;
/* Applicazione che verifica se un numero è primo */
class Primi{

    public static void main(String[] args){
        int n; //numero letto da input
        //pre: n>0
        /*lettura del numero da input*/
        System.out.println("Scrivi un numero (>0)");
        n = Lettore.in.leggiInt();
        if (primo(n))
            {System.out.print("il numero ");
             System.out.print(n);
             System.out.println(" è primo");}
        else
            {System.out.print("il numero ");
             System.out.print(n);
             System.out.println(" NON è primo");}
    }
}
```

# ... Applicazione per la verifica se un numero è primo

```
public static boolean primo(int n){
    int div; //divisori per la verifica
    boolean trovato; //true se è stato trovato un divisore

    div = 2;
    trovato = false;

    while (!trovato && (div<=n/2)){
        if (n%div == 0)
            trovato = true;
        div = div +1;}
    /* esco dal ciclo o quando trovato è true (il num è
       divisibile) o quando div > n/2 li ho provati tutti */

    return !trovato;

}
}
```

# Esercizi

- ❑ Scrivere un programma che generi i primi  $n$  numeri primi.
- ❑ Un numero  $n$  si dice perfetto se la somma di tutti i suoi divisori ( $n$  escluso) è uguale a  $n$ . Ad esempio 6 è perfetto perché  $1+2+3 = 6$ .  
Scrivere un'applicazione che legge da input un intero  $m$  e genera i primi  $m$  numeri perfetti.
- ❑ **Suggerimento:** non provare con valori di  $m$  troppo grandi, massimo  $m=4$  ....



# Riferimenti al libro di testo

---

- Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al **capitolo**
  - 5 su **Problemi, algoritmi e oggetti**
- Con particolare riferimento ai **paragrafi**
  - Dal 5.2.5 al 5.2.11