

# Laboratorio di Programmazione

## Laurea in Bioinformatica

25 gennaio 2010

### 1 Programmazione strutturata

#### 1.1 Esercizio 1

Si scriva lo pseudocodice di un programma che calcola la media di due numeri reali (si commenti l'esercizio nel caso che i due numeri siano interi).

#### 1.2 Esercizio 2

Si scriva lo pseudocodice di un programma che calcola la media di  $n$  valori  $x_1, \dots, x_n$ , in cui  $n$  è inserito dall'utente in fase di esecuzione del programma. Si ricordi che la media  $m$  di  $n$  valori  $x_1, \dots, x_n$  si calcola mediante la formula

$$m = \frac{1}{n} \sum_{i=1}^n x_i$$

Soluzione:

```
variabili n,i: numeri interi;
variabili m,x: numeri reali;
leggi n;
i = 0;
m = 0;
QUANDO i<n ESEGUI
    leggi x;
    m = m+x;
    i = i+1;
RIPETI
m = m/n;
scrivi m;
```

### 1.3 Esercizio 3

Si scriva lo pseudocodice per calcolare il valore massimo di una sequenza di 10 numeri interi inseriti dall'utente in fase di esecuzione.

Soluzione 1:

```
variabili max,x,i: numeri interi;
max = MINIMO_INT;
i = 0;
QUANDO i<10 ESEGUI
    leggi x;
    SE max < x;
        ALLORA
            max = x;
        FINESE
    i = i+1;
RIPETI
scrivi max;
```

N.B.: MINIMO\_INT e' l'intero piu' piccolo rappresentabile con il n. di bit a disposizione. Ad es., se gli interi sono rappresentati con 32 bit, MINIMO\_INT vale  $-2147483648$ .

Soluzione 2:

```
variabili max,x,i: numeri interi;
leggi x;
max = x;
i = 1;
QUANDO i<10 ESEGUI
    leggi x;
    SE max < x;
        ALLORA
            max = x;
        FINESE
    i = i+1;
RIPETI
scrivi max;
```

## 1.4 Esercizio 4

Si scriva lo pseudocodice per calcolare la differenza massima tra  $n$  numeri interi, con  $n$  inserito dall'utente in fase di esecuzione.

Soluzione:

```
variabili max,min,x,i,n: numeri interi;
max = MINIMO_INT;
min = MASSIMO_INT;
leggi n;
i = 0;
QUANDO i<n ESEGUI
    leggi x;
    SE max < x;
        ALLORA
            max = x;
    FINESE
    SE min > x;
        ALLORA
            min = x;
    FINESE
    i = i+1;
RIPETI
scrivi max-min;
```

## 2 JAVA: esercizi introduttivi

### 2.1 Esercizio 1

Si scriva il codice JAVA di un programma che quando eseguito visualizza il messaggio:

```
Ciao.  
Sono un programma scritto in Java.
```

Soluzione:

```
import java.io.*;  
  
class Saluto{  
    public static void main(String[] args){  
        System.out.print("Ciao.\n");  
        System.out.print("Sono un programma scritto in JAVA.\n");  
    }  
}
```

### 2.2 Esercizio 2

Si scriva il codice JAVA di un programma che calcola la media di due numeri interi. Si verifichi il funzionamento con le coppie di valori (3,7) e (4,7) e si cerchi una spiegazione per eventuali errori nel calcolo.

Soluzione:

```
import java.io.*;  
  
class mediaReali{  
    public static void main(String[] args) {  
  
        int x;  
        int y;  
        int media;  
        x = 3;  
        y = 7;  
  
        media = (x+y)/2;  
  
        System.out.println("Programma per il calcolo della media");  
        System.out.print("x = ");  
        System.out.println(x);  
        System.out.print("y = ");  
        System.out.println(y);  
        System.out.print("media = ");  
        System.out.println(media);  
    }  
}
```

### 2.3 Esercizio 3

Si scriva il codice JAVA di un programma areaTriangolo che calcola l'area di un triangolo di base 5 e altezza 10. Si ripeta l'esercizio nel caso di un triangolo di base 5 e altezza 7. Il tipo intero e' sufficiente in questo caso per rappresentare l'area?

Soluzione:

```

import java.io.*;

class areaTriangolo{
    public static void main(String[] args) {

        int baseTriangolo;
        int altezzaTriangolo;
        int areaTriangolo;
        baseTriangolo = 5;
        altezzaTriangolo = 10;

        areaTriangolo = (baseTriangolo*altezzaTriangolo)/2;

        System.out.println("Programma per il calcolo dell'area del triangolo");
        System.out.print("Base = ");
        System.out.println(baseTriangolo);
        System.out.print("Altezza = ");
        System.out.println(altezzaTriangolo);
        System.out.print("Area = ");
        System.out.println(areaTriangolo);
    }
}

```

## 2.4 Esercizio 4

Si modifichino gli esercizi 2 e 3 in modo che l'utente possa inserire in fase di esecuzione i valori (x,y) e (base,altezza).

Soluzione:

```

import prog.io.*;

class areaTriangolo{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        int baseTriangolo;
        int altezzaTriangolo;
        int areaTriangolo;

        System.out.print("Inserisci la base del triangolo: ");
        baseTriangolo = tastiera.readInt();
        System.out.print("Inserisci l'altezza del triangolo: ");
        altezzaTriangolo = tastiera.readInt();

        areaTriangolo = (baseTriangolo*altezzaTriangolo)/2; //att: troncamento ad intero

        System.out.println("Programma per il calcolo dell'area del triangolo");
        System.out.print("Base = ");
        System.out.println(baseTriangolo);
        System.out.print("Altezza = ");
        System.out.println(altezzaTriangolo);
        System.out.print("Area = ");
        System.out.println(areaTriangolo);
    }
}

```

## 2.5 Esercizio 5

Si scriva in JAVA un programma che accetta in input un numero intero n e stampa:

1. il resto della divisione per due
2. il valore  $2n$  se  $n$  è pari,  $3n$  se  $n$  è dispari

Il problema deve essere risolto senza far uso di costrutti di controllo.

Soluzione:

```
import prog.io.*;

class nPariDisp {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        out.print("Inserisci n: ");
        int n=in.readInt();

        int resto = n%2;

        int paridispari = n*(n%/2+2);

        out.print("il resto della divisione di n per due: ");
        out.println(resto);

        out.print("il valore 2n (per n pari) o 3n (per n dispari): ");
        out.println(paridispari);

    }
}
```

## 3 JAVA: esercizi introduttivi sulla classe String

### 3.1 Esercizio 2

Si scriva un programma che, dopo aver chiesto all'utente di inserire in tre passi successivi il giorno, il mese e l'anno, visualizza la data nel formato gg/mm/aaaa.

Soluzione:

```
import prog.io.*;

class Data{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String gg,mm,aaaa;

        out.print("Inserisci il giorno (gg): ");
        gg = tastiera.readLine();
        out.print("Inserisci il mese (mm): ");
        mm = tastiera.readLine();
        out.print("Inserisci l'anno (aaaa): ");
        aaaa = tastiera.readLine();

        out.print("La data immessa: "+gg+"/"+mm+"/"+aaaa);
    }
}
```

### 3.2 Esercizio 3

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa e due indici (numeri interi), stampa la sottostringa delimitata dai due indici e la sua lunghezza. Soluzione:

```
import prog.io.*;

class Sottostringa{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String stringa;
        int i1,i2;

        out.print("Inserisci una stringa: ");
        stringa = tastiera.readLine();
        out.print("Inserisci l'indice di inizio: ");
        i1 = tastiera.readInt();
        out.print("Inserisci l'indice di fine: ");
        i2 = tastiera.readInt();
        out.println("La sottostringa richiesta e': "+stringa.substring(i1,i2));
        out.println("La lunghezza della sottostringa e': "+stringa.substring(i1,i2).length());
    }
}
```

### 3.3 Esercizio 4

Si scriva un programma che, dopo aver chiesto all'utente di inserire due stringhe di cui la seconda e' una sottostringa della prima, stampa le lunghezze delle due stringhe e l'indice che individua la posizione della sottostringa nella prima stringa.

Soluzione:

```

import prog.io.*;

class Sottostringa2{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String stringa,substringa;

        System.out.print("Inserisci una stringa: ");
        stringa = tastiera.readLine();
        out.print("Inserisci una sottostringa della stringa precedente: ");
        substringa = tastiera.readLine();

        out.println("La lunghezza della stringa e': "+stringa.length());
        out.println("La lunghezza della sottostringa e': "+substringa.length());
        out.println("La sottostringa inizia alla posizione: "+stringa.indexOf(substringa));
    }
}

```

### 3.4 Esercizio 5

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa ed un carattere, stampa l'indice della prima occorrenza di quel carattere nella stringa e stampa una nuova stringa nella quale quel carattere e' sostituito dal carattere "X".

```

import prog.io.*;

class CharToX{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String stringa;
        String carattere;
        int posizione;

        out.print("Inserisci una stringa: ");
        stringa = tastiera.readLine();
        out.print("Inserisci un carattere: ");
        carattere = tastiera.readLine();

        posizione = stringa.indexOf(carattere);
        out.println("La nuova stringa e': "+stringa.substring(0,posizione)+
            "X"+stringa.substring(posizione+1,stringa.length()));
    }
}

```

## 4 JAVA: i costrutti di selezione "if/else" e iterazione "while"

### 4.1 Esercizio 1

Scrivere un programma che chiede all'utente di inserire i valori di due variabili intere var1 e var2 e, in base ad opportuni confronti fra le variabili, stampa la frase "Il valore di var1 è [uguale al | maggiore del | minore del] valore di var2".

```
import prog.io.*;

class confrontoStringhe{
    public static void main(String[] args){

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int var1, var2;

        out.println("Confronto fra interi ");
        out.print("Inserire il primo valore intero: ");
        var1 = in.readInt();
        out.print("Inserire il secondo valore intero: ");
        var2 = in.readInt();

        if(var1<var2)
            out.println("var1="+var1+" è minore di var2="+var2);
        else {
            if(var1>var2)
                out.println("var1="+var1+" è maggiore di var2="+var2);
            else {
                out.println("var1 e var2 sono uguali");
            }
        }
    }
}
```

### 4.2 Esercizio 2

Sul prezzo di un prodotto viene praticato lo sconto del 3% se costa meno di 500 euro e del 5% per prezzi superiori a 500 euro. Scrivere un programma che calcoli il prezzo da pagare.

### 4.3 Esercizio 3

Scrivere un programma che, lette due stringhe inserite dall'utente, indichi se sono uguali o diverse. Nel caso di stringhe diverse l'applicazione dovrà visualizzarle in ordine lessicografico e poi in ordine di lunghezza [es. 3.6 pag 80].

```
import prog.io.*;

class confrontoStringhe{
    public static void main(String[] args){

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String st1, st2;
        int lenst1, lenst2;

        out.println("Confronto di stringhe ");
        out.print("Inserire la prima stringa: ");
        st1 = in.readLine();
```

```

out.print("Inserire la seconda stringa: ");
st2 = in.readLine();

int cfr = st1.compareTo(st2);
if(cfr==0)
    out.println("Le due stringhe sono uguali");
else {
    lenst1=st1.length();
    lenst2=st2.length();
    out.println("Ordine lessicografico:");
    if(cfr<0) {
        out.println(st1);
        out.println(st2);}
    else {
        out.println(st2);
        out.println(st1);
    }
    out.println("Ordine di lunghezza:");
    if (lenst1==lenst2)
        out.println("Le due stringhe hanno stessa lunghezza");
    else if(lenst1<lenst2){
        out.println(st1);
        out.println(st2);}
    else {
        out.println(st2);
        out.println(st1);}
    }
}
}

```

## 4.4 Esercizio 4

La lunghezza della stringa di caratteri che esprime il numero 12 nella lingua italiana è 6, pari cioè alla metà del numero stesso. Utilizzando la classe `Intero` del package `prog.utili` scrivere una applicazione che legga un numero e verifichi se gode di questa proprietà. [Es. 3.7 p. 81]

## 4.5 Esercizio 5

Scrivere un programma per calcolare il valore massimo di una sequenza di numeri interi inseriti dall'utente in fase di esecuzione. Per gestire l'interazione con l'utente si usi una frase del tipo "Vuoi inserire un altro numero (s/n)?".

```

import prog.io.*;

class MaxDiN{
    public static void main(String[] args){

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String ans;
        int num, max;

        out.println("Massimo fra N numeri ");

        out.print("Inserisci il primo numero: ");
        num = in.readInt();
        max = num;

        out.println("Vuoi inserire un nuovo numero (s/n)? ");
        ans = in.readLine();

        while(ans.compareTo("s")==0){
            out.print("Inserisci il nuovo numero: ");
            num = in.readInt();

```

```

        if(num>max)
            max=num;

        out.println("Vuoi inserire un nuovo numero (s/n)? ");
        ans = in.readLine();
    }
    out.println("Ecco il massimo: "+max);
}
}

```

## 4.6 Esercizio 6

Si scriva un programma che, dopo aver chiesto all'utente di inserire una stringa ed un carattere, stampa l'indice di TUTTE le occorrenze di quel carattere nella stringa e stampa una nuova stringa nella quale TUTTE le occorrenze di quel carattere sono sostituite dal carattere "X". Si risolva il problema evitando l'uso del metodo "replace" della classe String.

Soluzione 1:

```

import prog.io.*;

class AllCharToX{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        String stringa, nuovastringa;
        String carattere;
        int posizione, posizioneprec;

        out.print("Inserisci una stringa: ");
        stringa = tastiera.readLine();
        out.print("Inserisci un carattere: ");
        carattere = tastiera.readLine();

        posizioneprec=0;
        posizione = stringa.indexOf(carattere);

        out.println("posizione: "+posizione);
        nuovastringa="";

        while(posizione>=0){
            nuovastringa=nuovastringa+stringa.substring(posizioneprec, (posizione+posizioneprec))+"X";
            posizioneprec=posizioneprec+posizione+1;
            out.println("posizioneprec: "+posizioneprec);

            posizione = stringa.substring(posizioneprec, stringa.length()).indexOf(carattere);
            out.println("posizione: "+posizione);
        }

        nuovastringa=nuovastringa+stringa.substring(posizioneprec, stringa.length());
        out.println("La nuova stringa e': "+nuovastringa);
    }
}

```

Soluzione 2:

```

import prog.io.*;

class AllCharToXver2{
    public static void main(String[] args) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
    }
}

```

```

String stringa;
String carattere;
int posizione;

out.print("Inserisci una stringa: ");
stringa = tastiera.readLine();
out.print("Inserisci un carattere: ");
carattere = tastiera.readLine();

posizione = stringa.indexOf(carattere);
out.println("posizione: "+posizione);

while(posizione>=0){
    stringa=stringa.substring(0,posizione)+"X"+stringa.substring(posizione+1,stringa.length());
    posizione = stringa.indexOf(carattere);

    out.println("posizione: "+posizione);
}
out.println("La nuova stringa e': "+stringa);
}
}

```

## 5 JAVA: il costrutto di iterazione "for"

### 5.1 Esercizio 1

Si scriva il programma RuotaStringa che:

1. acquisisce da standard input una stringa
2. produce una nuova stringa, la quale è una versione ruotata della stringa immessa. La rotazione avviene spostando tutti i simboli in base al valore assunto da un indice di rotazione, con la regola che simboli che uscirebbero dalla stringa rientrano dalla parte opposta:
  - un indice positivo n provoca la rotazione a destra di n posizioni della stringa
  - un indice negativo n provoca la rotazione a sinistra di n posizioni della stringaEsempio: n = 1 trasforma "remo" in "orem"  
Esempio: n = -1 trasforma "remo" in "emor"
3. stampa la stringa così ruotata

N.B.: e' obbligatorio l'utilizzo del costrutto "for" per la soluzione!

```
import prog.io.*;

class RuotaStringa {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        String iniziale = in.readLine("Inserisci stringa non vuota da ruotare: ");
        String finale = iniziale.toString(); // equivale a new String(iniziale.toString());

        int rotazione = in.readInt("Inserisci indice di rotazione: ");
        out.println("Rotazione...");

        if (rotazione>0)
            for (int r=0; r<rotazione; r++)
                finale = finale.substring(finale.length()-1) + finale.substring(0,finale.length()-1);
        else
            for (int r=0; r>rotazione; r--)
                finale = finale.substring(1) + finale.substring(0,1);

        out.println("Stringa ruotata: " + finale);
    }
}
```

### 5.2 Esercizio 2

Si scriva il programma TestPalindroma che verifica se la stringa immessa dall'utente è una parola palindroma. Esempi di parole palindrome sono: "ara", "anna", "otto", "ingegni", etc.

```
import prog.io.*;

class TestPalindroma {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        boolean palindroma=true;
        String stringa = in.readLine();
        //int substsx=0;
        //int substdx=stringa.length()-1;
```

```

    for (int sx=0, dx=stringa.length()-1 ; sx<=dx || palindroma==true ; sx++, dx--) {
        if(stringa.charAt(sx) != stringa.charAt(dx)) {
            palindroma=false;
        }
    }
    if(palindroma)
        out.println("La stringa \"" + stringa + "\"" e' palindroma.");
    else
        out.println("La stringa \"" + stringa + "\"" non e' palindroma.");
}
}

```

### 5.3 Esercizio 3

Si scriva il programma TutteLePalindrome che:

1. acquisisce dal file testuale "stringa.txt" una stringa
2. trova tutte le sottostringhe palindrome contenuta nella stringa d'ingresso  
Es.: in "1211123212221", la sottostringhe palindrome sono "121", "11", "111", "21112", "12321", etc.
3. stampa le sottostringhe trovate

N.B.: la soluzione riportata effettua il test di palindromicit  su sottostringhe di lunghezza  $\geq 2$ .

```

import prog.io.*;

class TutteLePalindrome {
    public static void main(String[] args) {

        FileInputManager inFile = new FileInputManager("stringa.txt");
        ConsoleOutputManager out = new ConsoleOutputManager();

        String stringa = inFile.readLine();

        out.println("Input da file: " + stringa);
        out.println("Ricerca delle sottostringhe palindrome...");

        for (int substsx=0; substsx<stringa.length()-1; substsx++){
            for (int substdx=stringa.length()-1; substdx>substsx; substdx--){

                /***** questo e' il solito check di palindromicita' *****/
                boolean palindroma=true;
                for (int sx=substsx, dx=substdx ; sx<=dx ; sx++, dx--) {
                    if(stringa.charAt(sx) != stringa.charAt(dx)) {
                        palindroma=false;
                    }
                }
                if(palindroma)
                    out.println("Sottostringa palindroma: " + stringa.substring(substsx,substdx+1));
                /*****
            }
        }
    }
}

```

### 5.4 Esercizio 4

Si scriva il programma CercaInHtml che, dato un file HTML e una parola da cercare, produce un nuovo file in cui tutte le occorrenze della parola sono evidenziate.

Suggerimenti:

1. in HTML, e' possibile evidenziare (ad es. in giallo) un frammento di testo usando il Tag "style" in questo modo:

<span style="background-color: #FFFF00">frammento di testo</span>

2. per gestire l'output su file usare la classe FileOutputManager della libreria prog.

```
import prog.io.*;

class CercaInHtml{
    public static void main(String[] args){

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        out.print("Indica il nome di un file .html: ");
        String nomefile = in.readLine();

        FileInputManager inFile = new FileInputManager(nomefile);

        String nuovohtml = "nuovo"+nomefile;
        FileOutputManager outFile = new FileOutputManager(nuovohtml);

        out.println("La parola da cercare?");
        String parola = in.readLine();

        out.println("Input da file: file.html");
        out.println("Ricerca della parola: "+parola);

        String rigadahtml = inFile.readLine(); //legge una riga del file
        while(rigadahtml!=null){

            String nuovarigadahtml=rigadahtml.replace(parola,
                "<span style=\"background-color: #FFFF00\">"+parola+"</span>");

            outFile.println(nuovarigadahtml);

            rigadahtml = inFile.readLine();
        }
        out.println("Il nuovo file html si chiama: "+nuovohtml);
        outFile.close();
    }
}
```

## 6 JAVA: uso di oggetti

### 6.1 Esercizio 1

Adoperando i metodi forniti dalla classe `Frazione` (cfr. pp. 72-73 del libro del corso) si scriva un programma java il quale:

1. Accetta da tastiera due coppie di valori interi positivi, creando due oggetti `Frazione` corrispondenti rispettivamente riferiti dalle variabili `f1` e `f2`
2. Calcola e stampa la differenza tra la frazione più grande e la più piccola e ne salva il riferimento in `f3` (N.B.: il risultato sarà per forza non negativo)
3. Calcola e stampa la divisione tra frazione più piccola e la più grande e ne salva il riferimento in `f4` (N.B.: il risultato sarà per forza minore di uno)
4. Copia l'oggetto riferito da `f4` in un nuovo oggetto `Frazione` riferito da `f5`
5. Verifica se gli ultimi due oggetti sono uguali oppure no e stampa quello riferito da `f5`
6. Verifica se gli ultimi due riferimenti sono uguali oppure no.

```
import prog.io.*;
import prog.utili.Frazione;

class FrazioneEs1 {
    static public void main (String args[]) {
        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager video = new ConsoleOutputManager();

        // Accetta da tastiera due coppie di valori interi positivi, creando due oggetti Frazione
        // corrispondenti rispettivamente riferiti dalle variabili f1 e f2
        Frazione f1 = new Frazione(tastiera.readInt("Numeratore Frazione 1: "),tastiera.readInt("Denominatore Frazione 1: "));
        Frazione f2 = new Frazione(tastiera.readInt("Numeratore Frazione 2: "),tastiera.readInt("Denominatore Frazione 2: "));

        // Calcola e stampa la differenza tra la frazione più grande e la più piccola e ne salva
        // il riferimento in f3 (N.B.: il risultato sarà per forza non negativo)
        Frazione f3;
        if (f1.isMaggiore(f2))
            f3 = f1.meno(f2);
        else
            f3 = f2.meno(f1);
        video.println("f3 = " + f3.toString());

        // Calcola e stampa la divisione tra frazione più piccola e la più grande e ne salva il
        // riferimento in f4 (N.B.: il risultato sarà per forza minore di uno)
        Frazione f4;
        if (f1.isMinore(f2))
            f4 = f1.diviso(f2);
        else
            f4 = f2.diviso(f1);
        video.println("f4 = " + f4.toString());

        // Copia l'oggetto riferito da f4 in un nuovo oggetto Frazione riferito da f5
        Frazione f5 = new Frazione(f4.getNumeratore(),f4.getDenominatore());

        // Verifica se gli ultimi due oggetti sono uguali oppure no e stampa quello riferito da f5
        if (f4.equals(f5))
            video.println("Gli oggetti riferiti da f4 e f5 sono uguali.");
        else
            video.println("Gli oggetti riferiti da f4 e f5 sono diversi.");
        video.println("f5 = " + f5.toString());

        // Verifica se gli ultimi due riferimenti sono uguali oppure no
        if (f4==f5)
            video.println("f4 e f5 sono uguali.");
        else
            video.println("f4 e f5 sono diversi.");
    }
}
```

## 6.2 Esercizio 2

Adoperando i metodi forniti dalla classe `Frazione` si scriva un programma java il quale:

1. Accetta da tastiera due valori interi positivi, creando un oggetto `Frazione` riferito dalla variabile `f1`
2. Calcola e stampa il reciproco della frazione appena data e ne salva il riferimento in `f2` (N.B.: qual è il reciproco di una frazione  $a/b$  ?)
3. Calcola e stampa la moltiplicazione tra le frazioni riferite da `f1` e `f2` e ne salva il riferimento in `f3` (N.B.: quanto deve valere questa moltiplicazione ?)
4. Calcola e stampa la divisione tra le frazioni riferite da `f1` e `f2` e ne salva il riferimento in `f4` (N.B.: quanto deve valere questa divisione ?)
5. Calcola il quadrato della frazione riferita da `f1` e ne salva il riferimento in `f5`
6. Verifica se gli oggetti riferiti da `f4` e `f5` sono uguali oppure no e stampa quello riferito da `f5`

```
import prog.io.*;
import prog.utili.Frazione;

class FrazioneEs2 {
    static public void main (String args[]) {

        ConsoleInputManager tastiera = new ConsoleInputManager();
        ConsoleOutputManager video = new ConsoleOutputManager();

        Frazione f1 = new Frazione(tastiera.readInt("Numeratore Frazione 1: "),tastiera.readInt("Denominatore Frazione 1: "));

        Frazione f2 = new Frazione(f1.getDenominatore(),f1.getNumeratore());
        video.println("f2 = " + f2.toString());

        Frazione f3 = f1.per(f2);
        video.println("f3 = " + f3.toString());

        Frazione f4 = f1.diviso(f2);
        video.println("f4 = " + f4.toString());

        Frazione f5 = f1.per(f1);
        video.println("f5 = " + f5.toString());

        if (f4.equals(f5))
            video.println("Gli oggetti riferiti da f4 e f5 sono uguali.");
        else
            video.println("Gli oggetti riferiti da f4 e f5 sono diversi.");
    }
}
```

# 7 JAVA: cast implicito ed esplicito, operatori postfissi e prefissi

## 7.1 Esercizio 1

Si scriva il programma TypeCast1 che:

1. dichiara tre variabili intere i,j, e k con valori rispettivamente 3,4, e 6
2. usando il cast di tipo implicito o esplicito, calcola la media corretta (rappresentata come float) in almeno 3 modi diversi e la media troncata (rappresentata con un intero) in almeno due modi diversi

```
import prog.io.*;

class TypeCast1 {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int i=3, j=4, k=6;
        float media_f;
        int media_i;

        media_f = (i+j+k)/3.0f;
        out.println("Media corretta: " + media_f);

        media_f = ((float)i+j+k)/3;
        out.println("Media corretta: " + media_f);

        media_f = (float)(i+j+k)/3;
        out.println("Media corretta: " + media_f);

        media_i = (i+j+k)/3;
        out.println("Media troncata: " + media_i);

        media_i = (int)((i+j+k)/3.0);
        out.println("Media troncata: " + media_i);
    }
}
```

## 7.2 Esercizio 2

Si scriva il programma TypeCast2 che:

1. legge da tastiera tre variabili intere i,j, e k
2. assegna la media dei tre numeri a una variabile del tipo strettamente necessario a evitare la perdita di informazioni

[suggerimento: si usi l'operatore % per le valutazioni sul tipo da usare]

```
import prog.io.*;

class TypeCast2 {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int i, j, k, somma;
        i=in.readInt("Inserisci il valore di i:");
        k=in.readInt("Inserisci il valore di j:");
        j=in.readInt("Inserisci il valore di k:");

        somma=i+j+k;
        if(somma%3 == 0){
            int media=somma/3;
            out.println("Media (int): " + media);
        }
        else {
            double media=somma/3.0;
            out.println("Media (double): " + media);
        }
    }
}
```

## 7.3 Esercizio 3

Si scriva il programma TypeCastRand che:

1. legge da tastiera il valore di un intero N
2. genera N numeri casuali interi compresi nell'intervallo [5,10]
3. ne calcola la media

[suggerimento: si usi la funzione Math.random() che genera numeri casuali di tipo double nell'intervallo [0,1] con distribuzione uniforme ]

```
import prog.io.*;

class TypeCastRand {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int sampleInt;
        int sumInt=0;
        int min=5, Max=10;
        double media;

        int N=in.readInt("Inserisci il n. di campioni: ");

        for(int i=0; i<N; i++){
            sampleInt=(int)(min+(Max-min)*Math.random());
            if(N<=5) out.println(sampleInt);
        }
    }
}
```

```

        sumInt+=sampleInt;
    }

    media = (double)sumInt/N;
    out.println("Media: "+media);
}
}
}

```

## 7.4 Esercizio 4

Il programma PrePost\_num sia scritto usando gli operatori pre e postfissi:

1. stampa il risultato di tre operazioni di incremento postfisso in sequenza (i messaggi a video siano del tipo "i=0", "stampa di i++: 0", "stampa di i++: 1", etc);
2. si ripeta la stessa cosa con l'incremento prefisso
3. si valutino le espressioni i=i++ e i=++i e si stampino a video dei messaggi del tipo ("i=0", "stampa di i=i++: ...", etc);
4. si valutino le espressioni i==i++ e i==++i come condizioni del costrutto IF e si stampi un messaggio che dica se sono sempre vere o sempre false

```

import prog.io.*;

class PrePost_num {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int i=0;
        out.println("i=0");
        out.println("stampa di i++: " + i++);
        out.println("stampa di i++: " + i++);
        out.println("stampa di i: " + i);
        out.println("*****");
        /*****/
        i=0;
        out.println("i=0");
        out.println("stampa di ++i: " + ++i);
        out.println("stampa di ++i: " + ++i);
        out.println("stampa di i: " + i);
        out.println("*****");
        /*****/
        i=0;
        out.println("i=0");
        out.println("stampa di i=i++: " + (i=i++));
        out.println("*****");

        i=0;
        out.println("i=0");
        out.println("stampa di i=++i: " + (i=++i));
        out.println("*****");
        /*****/
        if(i==i++)
            out.println("i==i++ e' sempre vera");
        else
            out.println("i==i++ e' sempre falsa");

        if(i==++i)
            out.println("i==++i e' sempre vera");
        else
            out.println("i==++i e' sempre falsa");
        out.println("*****");
    }
}

```

## 8 JAVA: il tipo char e il costrutto switch

### Esercizio Q2\_1

La classe CharCfr contenga un programma che:

1. Dichiarare due variabili c e d di tipo char e ne acquisisce i valori da tastiera;
2. A fronte di opportuni confronti stampa una o più frasi fra le seguenti:
  - "c e d contengono lo stesso carattere <valore di c,d> (codice Unicode: <unicode di c,d>)"
  - "c e d contengono caratteri differenti"
  - "nella codifica Unicode il carattere <c> (<unicode di c>) precede <d> (<unicode di d>)"
  - "nella codifica Unicode il carattere <c> (<unicode di c>) precede immediatamente <d> (<unicode di d>)"

```
import prog.io.*;

class CharCfr {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        char c,d;
        c=in.readChar("Inserire il primo carattere: ");
        d=in.readChar("Inserire il secondo carattere: ");

        if(c==d)
            out.println("c e d contengono lo stesso carattere "+c+" (codice Unicode: "+(int)c+" )");
        else {
            out.println("c e d contengono caratteri differenti");
            if(c==d-1) {
                out.println("e in Unicode "+c+" ("+(int)(c)+") precede immediatamente "+d+" ("+(int)(d)+")");
            } else {
                if(c<d) out.println("e in Unicode "+c+" ("+(int)(c)+") precede "+d+" ("+(int)(d)+")");
            }
        }
    }
}
```

## Esercizio Q2\_2

Si scriva un'applicazione per la simulazione di una semplice calcolatrice. Il programma deve scrivere il risultato dopo aver ricevuto da tastiera due numeri e il simbolo dell'operazione desiderata.

```
import prog.io.*;

class Calcolatrice {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int a=in.readInt("Inserire il primo operando: ");
        int b=in.readInt("Inserire il secondo operando: ");

        double risultato;
        char c=in.readChar("Inserire il tipo di operazione (+,-,*,/,%): ");

        switch (c) {
            case '+':
                risultato = a+b;
                out.println("Risultato: "+risultato);
                break;
            case '*':
                risultato = a*b;
                out.println("Risultato: "+risultato);
                break;
            case '-':
                risultato = a-b;
                out.println("Risultato: "+risultato);
                break;
            case '/':
                risultato = (double)a/b;
                out.println("Risultato: "+risultato);
                break;
            case '%':
                risultato = a%b;
                out.println("Risultato: "+risultato);
                break;
            default:
                out.println("operazione non valida");
        }
    }
}
```

## Esercizio Q2\_3

Si scriva un'applicazione che stampa tutte le lettere minuscole dell'alfabeto facendo uso dell'operatore postfisso di incremento. Nel codice appena scritto si sostituisca all'operatore postfisso l'operatore prefisso e si osservi il risultato.

```
import prog.io.*;

class StampaAlfab {
    public static void main(String[] args) {

        ConsoleOutputManager out = new ConsoleOutputManager();
        char i='a';
        do {
            out.print(i++ +" ");
        } while(i<='z');

        out.println("...");
    }
}
```

## Esercizio Q2\_4

Si scriva un'applicazione che permette di specificare somme di numeri interi inserendo da tastiera stringhe come "1+13+6+123".  
[sugg.: si ricorra ai metodi della classe String che permettono di individuare l'indice di un carattere (+ in questo caso) e di estrarre sottostringhe]

```
import prog.io.*;

class Calcolatrice2{
    public static void main(String[] args){

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        out.println("Somma di interi ");
        String st1;
        int fromIndex=0;
        st1 = in.readLine("Inserire degli interi separati dal +: ");
        int ind;
        String auxst;
        int somma=0;
        while((ind=st1.indexOf("+",fromIndex))>0){
            //out.println("trovato un piú in posizione: "+ind);
            auxst=st1.substring(fromIndex,ind);
            somma+=Integer.parseInt(auxst);
            fromIndex=ind+1;
        }

        auxst=st1.substring(fromIndex,st1.length());
        somma+=Integer.parseInt(auxst);
        out.println("Somma: "+somma);
    }
}
```

## 9 JAVA: metodi e campi statici

### Esercizio Q2\_5

Si scriva un programma per la risoluzione delle equazioni di secondo grado  
[sugg. si usi il metodo sqrt della class Math per il calcolo della radice quadrata]

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

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        double a,b,c;

        a=in.readDouble("Inserisci il coeff. a: ");
        b=in.readDouble("Inserisci il coeff. b: ");
        c=in.readDouble("Inserisci il coeff. c: ");

        double x1=(-b+Math.sqrt(b*b-4.0*a*c))/(2.0*a);
        double x2=(-b-Math.sqrt(b*b-4.0*a*c))/(2.0*a);
        out.println("x1: "+x1);
        out.println("x2: "+x2);
    }
}
```

### Esercizio Q2\_6

Si consideri un cerchio di raggio  $r$  il cui centro coincide con l'origine di una coppia di assi cartesiani. Si scriva un programma che:

1. generando una sequenza di  $N$  punti a caso di coordinate  $x$  e  $y$  comprese fra 0 ed  $r$ , si calcoli la frequenza dei punti che cadono nel quarto di cerchio di circonferenza (cioè il rapporto fra i punti che cadono nel quarto di circonferenza e il numero totale di punti generati). Per generare valori casuali si usi il metodo statico `Math.random()`.
2. usando il metodo al punto precedente calcolare un'approssimazione di  $\pi$  sapendo che la frequenza dei punti che cadono nel quarto di cerchio di circonferenza costituisce un'approssimazione del rapporto fra l'area  $A_q$  del quarto di cerchio e l'area del quadrato di lato  $r$ . Si ricordi poi che  $e' 4 * A_q = \pi * r^2$ .
3. si confronti il valore stimato di  $\pi$  con il valore della costante statica `Math.PI`.

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

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        double randx, randy;
        double r=2.5;
        int contatore=0;
        double freq;
        double stimaPI;

        int N=in.readInt("Inserisci il n. di campioni: ");

        for(int i=0; i<N; i++){

            randx=r*Math.random();
            randy=r*Math.random();

            if((randx*randx+randy*randy)<r*r)
                contatore+=1;
        }
    }
}
```

```

    freq = (double)contatore/N;
    out.println("Stima dell'area: "+((r*r)*freq*4.0));
    out.println("Area reale: "+(Math.PI*r*r));

    stimaPI = 4.0*freq;
    out.println("Stima di PI: "+stimaPI);
    out.println("PI in Math: "+Math.PI);
  }
}

```

## 10 Rappresentazioni numeriche

### 10.1 Esercizio 1

Si scriva un programma Java che converte un numero intero dalla rappresentazione in una base B generica a quella decimale. **Suggerimenti:** si usi il tipo stringa per l'input del numero da convertire; per la conversione da char a valore numerico di una cifra si può usare ad es. il metodo statico `int getNumericValue(char)` della classe `Character`.

```

import prog.io.*;
class ConvB10
{
    public static void main(String[] args)
    {
        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        double dec=0;
        out.print("Inserisci la base B");
        int B=in.readInt();
        out.print("Inserisci una stringa (in base B)");
        String c1=in.readLine();
        int len=c1.length();
        for (int i=0; i<len;i++){
            dec=dec+Character.getNumericValue(c1.charAt(i))*Math.pow(B,(len-i-1));
        }
        out.print("Il numero inserito corrisponde a ");
        out.print(dec);
    }
}

```

### 10.2 Esercizio 2

Si scriva un programma Java che converte un numero intero dalla rappresentazione decimale a quella in una base B generica.

### 10.3 Esercizio 3

Si scriva un programma che esegue la somma di due numeri binari in notazione complemento a 2 forniti dall'utente in fase di esecuzione. Il programma fornisca come risultato la rappresentazione decimale degli addendi e le rappresentazioni complemento a 2 e decimale della somma, segnalando possibili overflow.

## 11 JAVA: array e collezioni

### 11.1 Esercizio 1

Si scriva un programma che, dopo aver riempito un array di 5 interi con valori forniti dall'utente, lo stampa al contrario (partendo dall'ultimo elemento inserito, fino al primo).

### 11.2 Esercizio 2

Si scriva un programma che, dopo aver riempito un array di 5 interi con valori forniti dall'utente, copia gli elementi in ordine crescente in un secondo array e lo stampa.

## 11.3 Esercizio 3

La classe OrdinaStringhe contenga un programma che:

1. inizializza un array di stringhe acquisite da tastiera
2. ordina l'array per lunghezza crescente delle stringhe, adoperando una variabile di tipo stringa come contenitore temporaneo
3. stampa l'array così ordinato

```
import prog.io.*;

class OrdinaStringhe {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        final int MAX = 5;
        String[] arrStr = new String[MAX];
        String temp;

        for(int i=0; i<MAX; i++)
            arrStr[i] = in.readLine("Inserire la stringa n. " + (i+1) + ": ");

        for(int i=0; i<MAX; i++)
            for(int j=i+1; j<MAX; j++)
                if (arrStr[i].length() > arrStr[j].length()) {
                    temp = arrStr[i];
                    arrStr[i] = arrStr[j];
                    arrStr[j] = temp;
                }

        int counter = 1;
        for (String s : arrStr)
            out.println("Locazione " + counter++ + ": " + s);
    }
}
```

## 11.4 Esercizio 4

La classe OrdinaStringhe2 contenga un programma che:

1. inizializza un array di stringhe acquisite da tastiera
2. ordina gli elementi dell'array per lunghezza crescente delle stringhe in un nuovo array, senza adoperare alcun contenitore temporaneo
3. stampa il nuovo array

[SUGG.: copiare nel secondo array e poi escludere la stringa più corta del primo array via via fino a vuotare il primo array]

```
import prog.io.*;

class OrdinaStringhe2 {
    public static void main(String[] args) {

        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        final int MAX = 5;
        String[] arrStr = new String[MAX];
        String[] ordStr = new String[MAX];

        for(int i=0; i<MAX; i++)
            arrStr[i] = in.readLine("Inserire la stringa n. " + (i+1) + ": ");
```

```

int lunghMax;

for(int i=0; i<MAX; i++) {

    lunghMax = i;          // assumo la stringa più corta inizialmente in posizione i

    for(int j=i; j<MAX; j++) // scorro il vettore di partenza dalla posizione i
    if (arrStr[j].length() < arrStr[lungHMax].length())
        lunghMax = j;     // aggiorno la posizione della stringa più corta

    ordStr[i] = arrStr[lungHMax]; // copio il riferimento alla stringa più corta
    arrStr[lungHMax] = arrStr[i]; // sovrascrivo l'elemento copiato

    out.println("Locazione " + (i+1) + ": " + ordStr[i]);
}
}
}

```

## 12 JAVA: array di array

### Esercizio Q2\_9

La classe OpMatrici contenga un programma che:

1. crea due matrici 2x2, la prima con righe [1 2] e [3 4], la seconda con righe [5 6] e [7 8].
2. esegue la visita completa delle due matrici, prima per RIGA e poi per COLONNA, stampandone il risultato.
3. ne esegue il prodotto matriciale. Per verifica, si noti che il risultato corretto e' una matrice 2x2 con righe [19 22] e [43 50].

```
import prog.io.*;

class OpMatrici {
    public static void main(String[] args) {
        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int size1 = 2;
        int size2 = 2;
        int[][] Matr1 = {{1,2},{3,4}};
        int[][] Matr2 = {{5,6},{7,8}};

        out.println("====Visita per righe====");
        for(int i=0; i<size1; i++)
            for(int j=0; j<size2; j++)
                out.print(Matr1[i][j]+"\\t");
        for(int i=0; i<size1; i++)
            for(int j=0; j<size2; j++)
                out.print(Matr2[i][j]+"\\t");

        out.println("\\n====Visita per colonne====");
        for(int i=0; i<size2; i++)
            for(int j=0; j<size1; j++)
                out.print(Matr1[j][i]+"\\t");
        for(int i=0; i<size1; i++)
            for(int j=0; j<size2; j++)
                out.print(Matr2[j][i]+"\\t");

        out.println("\\n====Prodotto vettoriale====");
        int[][] ProdMatr=new int[size1][size2];
        for(int i=0; i<size1; i++){
            for(int j=0; j<size2; j++){
                ProdMatr[i][j]=0;
                for(int k=0; k<size2; k++)
                    ProdMatr[i][j] += Matr1[i][k]*Matr2[k][j];
                out.print(ProdMatr[i][j]+"\\t");
            }
            out.print("\\n");
        }
    }
}
```

## Esercizio Q2\_10

La classe LetturaVerticale contenga un programma che:

1. legge e salva in un array di stringhe una sequenza di stringhe da input fintantoché non viene immessa la stringa "FINE"
2. colloca le stringhe in una matrice di simboli avente un numero di colonne pari alla lunghezza della stringa minore, in cui la riga n-esima contiene i simboli della stringa n-esima
3. presenta a video una nuova sequenza, formata leggendo la matrice colonna dopo colonna.

Si risolva il problema prima adoperando un array di array di caratteri, poi lavorando col solo array di stringhe iniziale (ovvero senza creare la matrice di caratteri)

```
import prog.io.*;
class LetturaVerticale {
    public static void main(String[] args) {
        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();
        final int MAX = 1000;
        String[] aux = new String[MAX];
        String s;
        char c;
        int righe;

        for(righe=0; righe<MAX-1; righe++) {
            s = in.readLine("Immetti stringa n." + (righe+1) + " [FINE per terminare]: ");
            if(s.equals("FINE"))
                break;
            else
                aux[righe] = s;
        }

        String[] arrString = new String[righe];
        for(int i=0; i<righe; i++)
            arrString[i] = aux[i];
        int minimo = Integer.MAX_VALUE;

        for(String t : arrString)
            minimo = (t.length() < minimo ? t.length() : minimo);

        // Soluzione con array di array
        char[][] arrChar = new char[righe][minimo];
        for (int i=0; i<righe; i++) // crea matrice
            for (int j=0; j<minimo; j++)
                arrChar[i][j] = arrString[i].charAt(j);

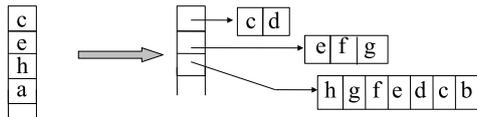
        for (int j=0; j<minimo; j++) { // stampa matrice
            for (int i=0; i<righe; i++)
                out.print(arrChar[i][j]);
            out.println();
        }

        // Soluzione con array di stringhe
        for (int j=0; j<minimo; j++) { // stampa array
            for (int i=0; i<righe; i++)
                out.print(arrString[i].charAt(j));
            out.println();
        }
    }
}
```

## Esercizio Q2\_11

La classe CaratteriNelMezzo contenga un programma che:

1. inizializza un array di 10 simboli UNICODE scelti a caso tra i codici decimali 1 e 150 adoperando il metodo statico `random()` di `Math`
2. sostituisce col carattere 'a' gli elementi nell'array che non sono alfabetici minuscoli
3. inserisce carattere 'z' nell'ultimo elemento dell'array, e infine stampa gli elementi dell'array risultante
4. per ogni elemento dell'array escluso l'ultimo, costruisce un nuovo array che contiene tutti i caratteri intermedi tra quello contenuto in quell'elemento (compreso) e quello contenuto nell'elemento di indice superiore (escluso), in avanti o all'indietro a seconda che l'elemento di indice superiore contenga un carattere rispettivamente posteriore o anteriore nell'ordinamento fissato dal codice UNICODE. Ad esempio:



5. stampa, riga dopo riga, ciascuna lista di caratteri contenuti nei nuovi array così definiti

```
import prog.io.*;
class CaratteriNelMezzo {
    public static void main(String[] args) {
        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        final int MAX = 10;
        char[] arrChar = new char[MAX];
        int distanza;

        for(int i=0; i<MAX-1; i++) {
            arrChar[i] = (char)(1+ Math.random()*150);
            if(arrChar[i]<'a' || arrChar[i]>'z')
                arrChar[i] = 'a';
        }

        arrChar[MAX-1] = 'z';
        for (char c : arrChar)
            out.println(c);

        char[][] arrMezzo = new char[MAX-1][];

        for(int i=0; i<MAX-1; i++)
            if(arrChar[i+1]-arrChar[i] >0) {
                distanza = arrChar[i+1]-arrChar[i];
                arrMezzo[i] = new char[distanza];
                for(int j=0; j<distanza; j++)
                    arrMezzo[i][j] = (char)(arrChar[i] + j);
            }
            else {
                distanza = arrChar[i]-arrChar[i+1];
                arrMezzo[i] = new char[distanza];
                for(int j=0; j<distanza; j++)
                    arrMezzo[i][j] = (char)(arrChar[i] - j);
            }

        for(int i=0; i<MAX-1; i++) {
            for (char c : arrMezzo[i])
                out.print(c);
            out.println();
        }
    }
}
```