Corso di Laurea in Matematica
Corsi di Informatica (n.o.) e Informatica 2 (v.o.)

UNIVERSITÀ DELLA CALABRIA
Dipartimento di Matematica e Informatica

Prova d'Esame del 24/06/2016

Esercizio 1. Si implementi in Java un metodo "booleano" (che cioè restituisca "vero" o "falso") che riceve come parametro un array di caratteri, sia "v". Il metodo deve leggere da standard input una sequenza di caratteri e trascrivere parte dei caratteri letti nell'array "v" ricevuto come parametro, secondo quanto descritto di seguito.

- La sequenza contiene dei caratteri qualunque, comprese parentesi tonde (aperte e chiuse).
- Si può supporre che non ci siano parentesi "innestate": questo vuol dire che si può assumere che, dopo aver incontrato una parentesi aperta, non se ne incontrerà un'altra prima di aver incontrato una parentesi chiusa; inoltre, ogni parentesi aperta avrà la propria corrispondente chiusa. In altre parole, ogni parentesi tonda aperta deve essere chiusa, prima di aprire una nuova parentesi e comunque prima di terminare la sequenza.
- La sequenza è terminata da un "tappo" costituito dal carattere "$".
- Il metodo deve copiare nell'array "v" le sottosequenze di caratteri contenute tra parentesi, nello stesso ordine in cui le incontra in input.
- Se l'array si "riempie" prima che sia terminata la sequenza (cioè prima che in input si incontri il tappo), il metodo deve interrompere la lettura e restituire "false" (nota: non è necessario modificare gli elementi già copiati in "v"); se invece la sequenza termina prima che l'array sia "pieno", il metodo deve inserire nella prima posizione libera dell'array stesso (in altre parole la posizione successiva all'ultimo carattere copiato) il carattere "#" e restituire "true".

Esempio: se la sequenza fosse

(**s d f**) d t (**g**) t y * 6 ( ) k (**r t y 5**) $

le sottosequenze da "copiare" nell'array "v" sarebbero 4: "sdf", "g", "", "rty5". Si noti come la terza sottosequenza sia vuota, e quindi non comporterà la scrittura di alcun carattere in "v". Pertanto, il contenuto dell'array "v", nel caso fosse di dimensione 10, dovrebbe essere

s	d	f	g	r	T	y	5	#	

Nel caso sopra descritto, la dimensione dell'array "v" è sufficiente, quindi può essere riempito come illustrato e il metodo dovrà restituire "true". Se la dimensione di "v", invece, fosse minore. e perciò insufficiente, ad esempio pari a 6 o pari a 8, questo dovrà essere stato riempito come di seguito illustrato

s	d	f	g	r	t

s	d	f	g	r	t	y	5

e il metodo dovrà restituire "false".

SOLUZIONE:

```
public static boolean leggiCopia(char[] v){

            Scanner in=new Scanner(System.in);
            System.out.println("in");
            char c = in.next().charAt(0);
            boolean inParentesi=false;
            int i=0;

            while(c!='$'){
                  // se v e' pieno posso restituire false
                  // perche' la sequenza non e' ancora terminata
                  if(i>=v.length){
                        return false;
                  }

                  if(inParentesi){//se sono tra due parentesi
                        if(c==')'){
                              //chiusa parentesi
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 24/06/2016

```
                        inParentesi=false;
                    }
                    else{
                        //scrivo il carattere in v e incremento indice
                        v[i]=c;
                        i++;
                    }
                }
                else{//se non sono tra due parentesi

                    if(c=='('){//mi interessa solo la parentesi aperta
                        inParentesi=true;
                    }
                }
                //leggo il prossimo carattere
                System.out.println("Inserisci il prossimo carattere o il tappo $");
                c = in.next().charAt(0);
            }

            //se non c'e' lo spazio per mettere il carattere #
            if(i>=v.length){
                return false;
            }
            else{
                //tutto ok
                v[i]='#';
                return true;
            }
        }
```

Esercizio 2. Si implementi in Java un metodo "void" che, ricevuti come parametri una matrice di interi (sia essa *M*) e due array di interi (siano essi "inp" e "out"), si comporti come descritto di seguito.

- Per ogni numero PARI contenuto in "inp", si devono cercare tutte le RIGHE di M che contengono quel numero (se ce ne sono), e copiare gli indici di queste righe in "out".
- Per ogni numero DISPARI contenuto in "inp", si devono cercare tutte le COLONNE di M che contengono quel numero (se ce ne sono), e copiare gli indici di queste righe in "out".
- Si può supporre che gli array siano tutti già creati, e che le dimensioni di "out" siano SEMPRE sufficienti a contenere gli indici indicati.
- Non importa l'ordine in cui i numeri corrispondenti agli indici di riga/colonna sono copiati in "out".

Esempio:
Se in input avessimo la seguente matrice M (3 X 5) e il seguente array "inp" (di dimensione 4):

	0	*1*	*2*	*3*	*4*	*5*
0	**2**	1	**5**	6	4	**2**
1	3	**7**	**2**	**5**	4	3
2	**2**	1	**7**	**8**	3	3

2	7	5	8

l'array "out" dovrebbe essere riempito con gli indici seguenti: **0 0 1 2 2 1 2 2 3**, **in qualunque ordine**. Infatti, le righe 0, 1 e 2 contengono il "2" (due occorrenze in riga zero) e/o l'"8" (pari); le colonne 1, 2 e 3, invece, contengono il 7 e/o il 5 (dispari). L'array potrebbe essere riempito anche in altro ordine.

SOLUZIONE:

CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 24/06/2016

```
public static void pariDispariVersione1(int[][] M, int[] inp, int[] out) {

            int indexOut = 0;
            for (int i = 0; i < inp.length; i++) {// itero sui numeri da cercare
                  int daTrovare = inp[i];

                  for (int riga = 0; riga < M.length; riga++) {
                        for (int col = 0; col < M[riga].length; col++) {
                              if (daTrovare == M[riga][col]) {

                                    if(daTrovare%2==0){
                                    //numero pari, aggiungo l'indice di riga
                                          out[indexOut]=riga;
                                    }
                                    else{
                                    //numero dispari, aggiungo l'indice di colonna
                                          out[indexOut]=col;
                                    }
                                    indexOut++;//incremento l'indice per out
                              }
                        }
                  }
            }
      }

// versione alternativa
public static void pariDispariVersione2(int[][] M, int[] inp, int[] out) {
            int indexOut = 0;

            //itero su tutti i numeri della matrice
            for (int riga = 0; riga < M.length; riga++) {
                  for (int col = 0; col < M[riga].length; col++) {

                        int daTrovare = M[riga][col];
                        //cerco il numero in inp
                        for (int i = 0; i < inp.length; i++) {

                              if(daTrovare==inp[i]){ //trovato

                                    if(daTrovare%2==0){
                                    //numero pari, aggiungo l'indice di riga
                                          out[indexOut]=riga;
                                    }
                                    else{
                                    //numero dispari, aggiungo l'indice di colonna
                                          out[indexOut]=col;
                                    }
                                    indexOut++;//incremento l'indice per out
                              }
                        }
                  }
            }

      }
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it