

Due Esercizi su Ricorsione (“J, K”)

(Es.J) Usando il seguente metodo:

```
//inserisco copia degli elementi di daArr negli elementi di
  adArr
//scrivendoli in adArr a partire dal suo indice [daPos]
public static void copiaArrInPos(int[] daArr, int[] adArr,
  int daPos){
//assumo che daArr.lenght+daPos <= adArr.length (altrimenti
  ho fault)
  for(int i=0; i<daArr.length; i++)
    adArr[daPos+i] = daArr[i];
}
```

e l'espressione `(int) (Math.random()*100)`, che crea un intero casuale, scrivere (e invocare opportunamente da main) un metodo RICORSIVO:

```
public static int[] creaArrPalCas(int numCoppie)
```

che crea e restituisce un array di interi palindromo, con un numero di elementi pari a $2 \cdot \text{numCoppie}$, dove gli elementi sono casuali e ripetuti a coppie come in: 9 8 8 9, oppure 1 2 3 4 3 2 1 (palindromi).

L'array palindromo richiesto si può costruire con ricorsione osservando che:

- palindromo di (n) coppie = num.x | palindromo di (n-1) coppie | num.x

(Es.K) scrivere (e invocare opportunamente da main) un metodo RICORSIVO:

```
public static boolean isArrPal(String[] arrStr)
```

che riceve un array di stringhe e ritorna true solo se l'array ricevuto è palindromo al suo interno, cioè le stringhe nei suoi elementi sono uguali a 2 a 2, come: "x" "x", oppure "a" "b" "a".

La verifica si può effettuare con ricorsione confrontando gli elementi della coppia più esterna e invocando poi il metodo stesso sulla sezione interna dell'array.