

E20 – Esercizi sulla ricorsione

Esercizio 1. Si scriva un metodo ricorsivo che calcola il coefficiente binomiale $\binom{n}{k}$ (per $n, k \in \mathbb{N}$ e $0 \leq k \leq n$) che può essere definito induttivamente come segue.

$$\binom{n}{k} = \begin{cases} 1 & \text{se } n = 0 \text{ o } k = 0 \\ \binom{n-1}{k} + \binom{n-1}{k-1} & \text{se } n > 0 \text{ e } k > 0 \end{cases}$$

Esercizio 2. Scrivere un metodo di classe **ricorsivo** che riceve in input due stringhe **s1** e **s2** ed altri parametri se necessario e restituisce true se le due stringhe sono una l'inverso dell'altra (cioè **s1** letta da destra verso sinistra è uguale a **s2**).

Esercizio 3. Scrivere un metodo di classe **ricorsivo** che riceve in input una matrice quadrata di interi **a**, due interi **i** e **j** ed altri parametri se necessario, e restituisce true se gli elementi della riga **i**-esima e della colonna **j**-esima sono tutti uguali a 0.

Esercizio 4. Sia data la classe Persona mostrata qui di seguito.

```
public class Persona{
    private String nome;
    private Persona padre;
    private Persona madre;

    public Persona(String nome, Persona padre, Persona madre){
        this.nome=nome;
        this.padre=padre;
        this.madre=madre;
    }

    public String getNome(){
        return this.nome;
    }

    public Persona getPadre (){
        return this.padre;
    }

    public Persona getMadre (){
        return this.madre;
    }
}
```

L'insieme degli antenati di una persona può essere definito come segue:

- Se il padre e la madre non sono noti, l'insieme degli antenati è vuoto.
- Se il padre è noto ma la madre no, l'insieme degli antenati è costituito dal padre e dai suoi antenati.
- Se il padre non è noto e la madre sì, l'insieme degli antenati è costituito dalla madre e dai suoi antenati.
- Se entrambi i genitori sono noti, l'insieme degli antenati è costituito dal padre, dalla madre e dai loro antenati.

Scrivere un metodo ricorsivo che data una Persona p stampi tutti gli antenati di p.

Soluzioni

Esercizio 1 – svolgimento.

```
public static int binomiale(int n, int k){
    int b;

    if(n==0 || k==0)
        b=1;
    else{
        b=binomiale(n-1,k)+binomiale(n-1,k-1);
    }

    return b;
}
```

Esercizio 2 – svolgimento.

```
public static boolean inverse(String s1, String s2, int i){
    boolean inv;
    int n1=s1.length();
    int n2=s2.length();

    if(i==n1 && i==n2){
        inv=true;
    }else if((i==n1 && i!=n2)|| (i!=n1 && i==n2)){
        inv=false;
    }else{
        inv=(s1.charAt(i)==s2.charAt(n2-1-i)) && inverse(s1,s2,i+1);
    }
    return inv;
}

public static boolean inverse(String s1, String s2){
    return inverse(s1,s2,0);
}
```

Esercizio 3 – svolgimento.

```
public static boolean rigaEColonnaUno(int[][] a, int i, int j, int k){
    boolean risultato;
    if(k==a.length){
        risultato=true;
    }else{
        risultato=(a[i][k]==1 && a[k][j]==1) && rigaEColonnaUno(a,i,j,k+1);
    }

    return risultato;
}

public static boolean rigaEColonnaUno(int[][] a, int i, int j){
    return rigaEColonnaUno(a, i, j, 0);
}
```

Esercizio 4 – svolgimento.

```
public static void stampaAntenati(Persona p){
    if(p.getPadre()!=null){
        System.out.println(p.getPadre().getNome());
        stampaAntenati(p.getPadre());
    }
    if(p.getMadre()!=null){
        System.out.println(p.getMadre().getNome());
        stampaAntenati(p.getMadre());
    }
}
```