

```

/*
 * Prof. Mauro De Berardis 2020
 * Il programma, implementato in vari linguaggi di programmazione e qui proposto in Java, visualizza le combinazioni di n bit,
 * con n fornito in input e compreso tra 1 e 24. Viene calcolato il tempo di esecuzione.
 * L'algoritmo è quello utilizzato quando l'esercizio fu proposto nell'anno scolastico 2011-2012 agli alunni della classe IVBI
 * dell'ITS Alessandrini di Teramo e allora implementato in C++:
 * 1. si fornisce il valore di n
 * 2. si calcola il numero di combinazioni k pari a 2^n
 * 3. ciascun numero decimale da 0 a k-1 viene convertito in una stringa binaria che viene formattata aggiungendo uno o più zeri
 *    iniziali in modo tale che la lunghezza sia sempre uguale ad n
 * Ad esempio: n=3 Numero di combinazioni: 2^3=8 Si convertono in binario i numeri decimali da 0 a 7 e si formattano a lunghezza 3
 * Risultato: 000 001 010 011 100 101 110 111
 */
import java.io.BufferedReader;
import java.io.InputStreamReader;

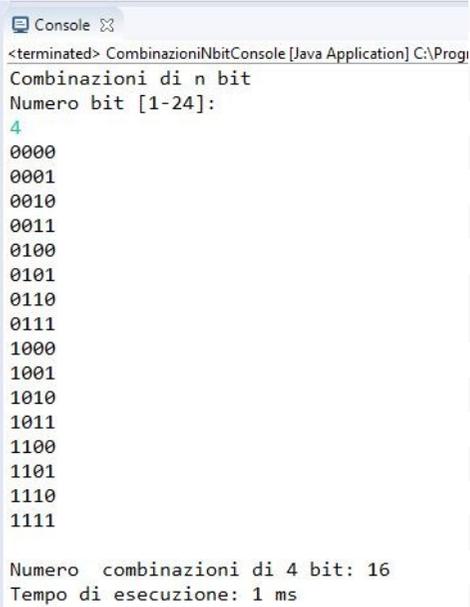
public class CombinazioniNbitConsole {

    static String formatta(String s, int l)
    {
        String zeri="000000000000000000000000";
        return zeri.substring(0,l-s.length()+s;
        // 24 zeri utili per inserire gli zeri iniziali
    }

    public static void main(String[] args) {

        System.out.println("Combinazioni di n bit ");
        InputStreamReader input=new InputStreamReader(System.in);
        BufferedReader tastiera=new BufferedReader(input);
        String valoreletto;
        int n;
        do {
            try
            {
                System.out.println("Numero bit [1-24]: ");
                valoreletto=tastiera.readLine();
                n=Integer.parseInt(valoreletto);
            }

```



```

Console
<terminated> CombinazioniNbitConsole [Java Application] C:\Progr
Combinazioni di n bit
Numero bit [1-24]:
4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
Numero combinazioni di 4 bit: 16
Tempo di esecuzione: 1 ms

```

```
    catch(Exception e)
    {
        System.out.println("Errore");return;
    }

}while(n<1 || n>24);
long startTime = System.currentTimeMillis();

double ncombinazioni=Math.pow(2.0, n);
for(int i=0;i<ncombinazioni;i++)
{
    String binario=formatta(Integer.toString(i, 2),n);
    System.out.println(binario);
}
long endTime = System.currentTimeMillis();
long elapsedTime = endTime- startTime;
long nc=(long)ncombinazioni;
System.out.println("\nNumero combinazioni di "+String.valueOf(n)+" bit: "+String.valueOf(nc));

if(elapsedTime<1) {
    System.out.println("Tempo di esecuzione: < 1 ms");
}else {
    System.out.println("Tempo di esecuzione: "+String.valueOf(elapsedTime)+ " ms");
}
System.exit(0);
}
```