Exercício 1: Algoritmo para encontrar *k*-ésimo numero natural par.

Exercício 1: Algoritmo para encontar k-ésimo numero natural par.

Algoritmo 1: Even(positive integer ***k*)**
Input: ***k*** , a positive integer
Output: ***k***-th even natural number (the first even being ***0***)

Algorithm:
if *k* = *1*, then return ***0*;**
else return **Even(*k*-*1*) + *2* .**

Exercício 2: Algoritmo para calcular 2^k

Exercício 2: Algoritmo para calcular 2^k

Algorithm 2 Power_of_2(natural number **k)**
Input: ***k*** , a natural number
Output: ***k***-th power of ***2***

Algorithm:
if ***k* = *0***, **then** return ***1***;
else return **2*Power_of_2(*k* - *1*) .**

Exercício 3: Algoritmo recursivo para busca seqüencial.

Exercício 3: Algoritmo recursivo para busca seqüencial.

Algorithm 3 SeqSearch(*L, i, j, x*)

Input: ***L*** is an array, ***i*** and ***j*** are positive integers, ***i*** $\leq$ ***j***, and ***x*** is the key to be searched for in ***L***.
Output: If ***x*** is in ***L*** between indexes ***i*** and ***j***, then output its index, else output ***0***.

Algorithm:
if *i* $\leq$ ***j* , then**
{
  if *L*(*i*) = *x*, then return ***i* ;**
  else return **SeqSearch(*L, i+1, j, x*)**
}
else return ***0*.**

Exercício 4: Palindrome Checker

Exercício 4: Palindrome Checker

```
Pal(str: String of length n)
{
      if (n==0 or n==1)
      {
            return true;
      }
      else
      {
            if (Pal(substring of str excluding first and last character)
                        and (character_1==character_n))
            {
                  return true;
            }
            else
            {
                  return false;
            }
      }
}
```

Exercício 5: Algoritmo recursivo para multiplicação de números naturais

Exercício 5: Algoritmo recursivo para multiplicação de números naturais

a*b = a se b = 1

a*b = a*(b-1)+a se b >1

Exercício 6: Imagine *v* como um vetor de inteiros. Apresente algoritmos recursivos para calcular:

1) O elemento máximo do vetor;
2) A soma dos elementos do vetor;
3) A media dos elementos do vetor.

Exercício 7: Conta numero de primo entre *a* e *b*.

Exercício 8: Conta numero de primo entre *a* e *b*.

$$count\_prime(a,b) = \begin{cases} 0, & se \quad a > b \\ count\_prime(a, b-1) + b, & se\ prime(b) \\ count\_prime(a, b-1), & \text{sen}\,ão \end{cases}$$

Exercício 9: A função de Ackerman é definida recursivamente sobre os inteiros não-negativos como segue:

a(m, n) = n+1, se m = 0

a(m, n) = a(m-1, 1), se m != 0 e n = 0

a(m, n) = a(m-1, a(m, n-1)) se m != 0, n != 0

1) Demonstre que a(2, 2) = 7
2) Você consegue descobrir um método iterativo para calcular a(m, n)?