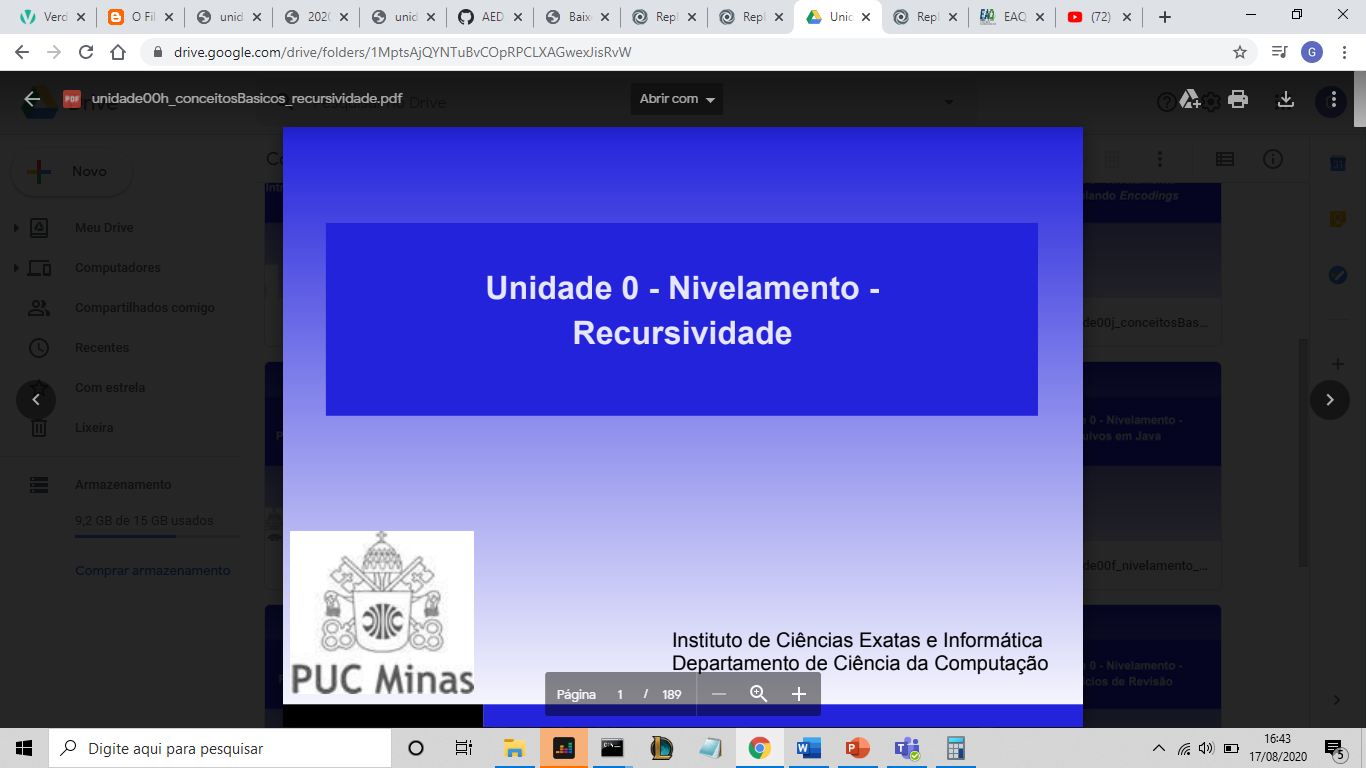
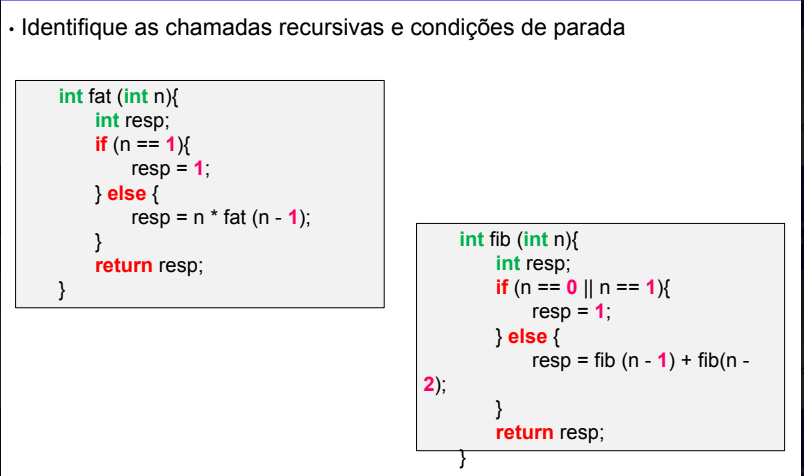
Trabalho Teórico 1



SLIDE H



* Chamadas recursivas: resp = fib(n-1)+fib(n-2)

resp =n \*fat(n-1)

* Condição de parada: if(n==1)

If(n==0 || n==1)



void mostrar ()

{

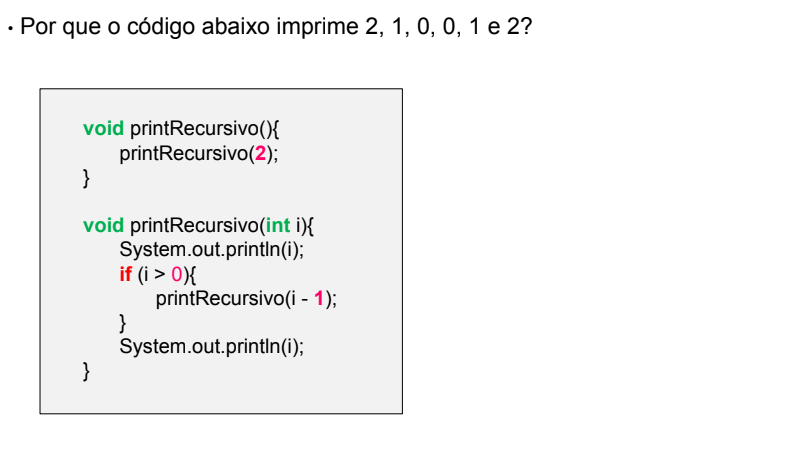
for (int i = 0; i < 4; i = i + 1)

{

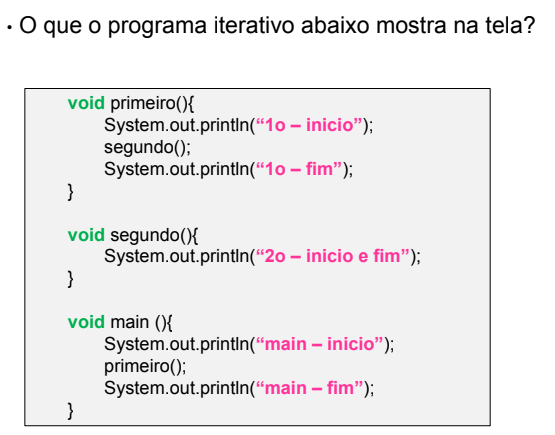
System.out.println(i);

}

}



O código imprime o 2 e chama a função recursiva passando o parâmetro (i-1) 2-1.Logo após a condição de parada ser executa i>0 a função vai retornar para função que a chamou onde o i = 1 e retornara de novo onde o i=2.



1º| main-inicio

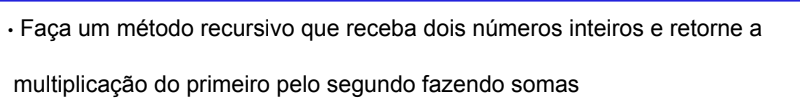
2º|1o – início

3°|2o – início e fim

4º|1o – fim

5º|2o – início e fim

6°|main – fim



#include <stdio.h>

int multiplica(int x, int y){

int resp=0;

if(y>0)

resp = x + (multiplica(x, y-1));

return resp;

}

int main(void) {

int resp=0, x=2, y=9;

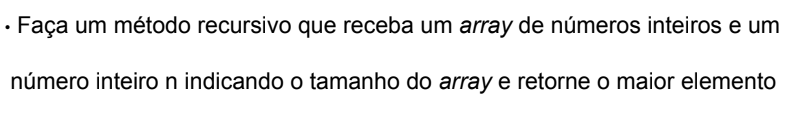
resp=multiplica(x,y);

printf("%d",resp);

return 0;

}

<https://repl.it/join/hmblrvar-guilhermecossoc>



#include <stdio.h>

int maior(int vet[], int n)

{

return maiore(vet , n, 0);

}

int maiore(int vet[], int n , int i)

{

int resp;

if(n==n-1)

resp = vet[n-1];

else

{

resp = maiore(vet, n, i+1);

if(resp<vet[i])

resp=vet[i];

}

return resp;

}

int main(void) {

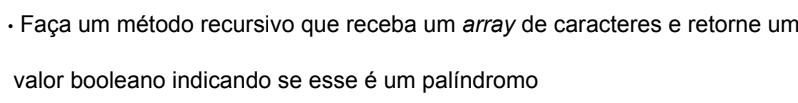
int vetor[10]={1,2,3,4,5,15,7,8,9,10};

int tam=10;

int resp = maior(vetor, tam);

printf("Resposta = %d", resp);

}



import java.io.File;

Palindromo

{

  public static boolean palin(String s)

  {

    boolean resp=true;

    if(s.length()==class 2 )

      if(s.charAt(0) != s.charAt(1))

        resp=false;

    for(int i=0 ;i<s.length()/2 ;i++)

    {

      if(s.charAt(i)!=s.charAt(s.length()-1-i))

      {

        resp=false;

        return resp;

      }

    }

    return resp;

  }

  public static void main(String[] args){

    boolean resp=false;

    String str=MyIO.readLine();

   while(!str.contains("FIM"))

     {

             resp=palin(str);

       if(resp==true)

           MyIO.println("SIM");

else

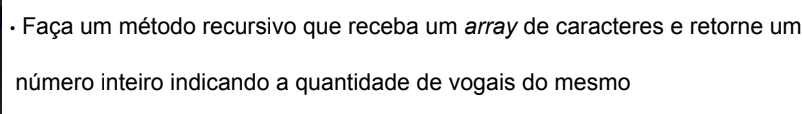
           MyIO.println("NAO");

   str=MyIO.readLine();

}

  }

}



class Isrecursiva

{

  public static int Vogal(String s)

  {

    s = s.toUpperCase();

    return Vogal(s, 0 );

  }

  public static int Vogal(String s,int i)

  {

    int resp = 0;

    if(i<s.length())

  {

      if(s.charAt(i) == 'A' || s.charAt(i) == 'E' || s.charAt(i) == 'I' || s.charAt(i) == 'O' || s.charAt(i) == 'U' )

resp ++;

          resp += Vogal(s, i + 1);

      else

          resp = 0;

    }

    return resp;

  }

  public static void main(String[] args)

  {

    String str=MyIO.readLine();

    int vogal;

   while(!str.contains("FIM"))

    {

        vogal=Vogal(str);

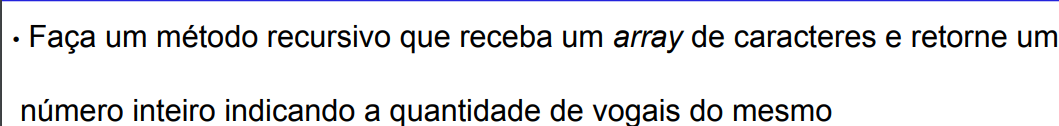
        print( vogal );

   str=MyIO.readLine();

    }

  }

}



class Qttvogal

{

  public static int Vogal(String s)

  {

    s = s.toUpperCase();

    int resp = 0;

    return Vogal(s, 0,resp );

  }

  public static int Vogal(String s,int i,int resp)

  {

    if(i<s.length())

  {

      if(s.charAt(i) == 'A' || s.charAt(i) == 'E' || s.charAt(i) == 'I' || s.charAt(i) == 'O' || s.charAt(i) == 'U' )

          resp = Vogal(s, i + 1, resp+1);

    }

    return resp;

  }

  public static void main(String[] args)

  {

    String str=MyIO.readLine();

    int vogal;

   while(!str.contains("FIM"))

    {

        vogal=Vogal(str);

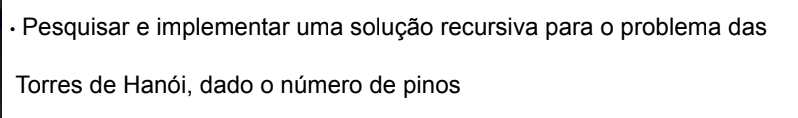
        MyIO.print( vogal);

        str=MyIO.readLine();

    }

  }

}



Solução

O numero de movimentos se da pela seguinte formula :

Numero de pinos mim=3

mim =7

mim=mim\*2+1;