Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação I
AP1 1^o **semestre de 2016.**

---

1 Primeira questão (2 pontos)

Defina em um parágrafo (meia dúzia de frases) o que é programação dinâmica.

Programação dinâmica é uma técnica para solução de problemas complexos pela sua subdivisão em um conjunto de subproblemas mais simples, que são resolvidos apenas uma vez, e cujas soluções são armazenadas em alguma estrutura de dados em memória. Na próxima vez que o mesmo subproblema aparecer, ao invés de recomputar sua solução, basta consultar a solução computada previamente.

2 Segunda questão (2 pontos)

Desenhe uma interface para o lazarus que permita ler um valor n, exibir os n primeiros elementos da sequência de Fibonacci, e o número de chamadas recursivas executadas. Forneça os nomes de todos os componentes utilizados.

Sequência de Fibonacci:
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...

3 Terceira questão (4 pontos)

Considere um array de inteiros longos, **fibs**.

1. Complete o procedimento **recursivo** *fibo* abaixo (3-5 linhas estão faltando em (2)), para que retorne o n-ésimo elemento da sequência de Fibonacci, realizando apenas n chamadas recursivas.

2. Qual o conteúdo esperado para o array fibs, após a chamada **fibo(5)**?

   fibs[0] = 1, fibs[1] = 1, fibs[2] = 2, fibs[3] = 3, fibs[4] = 5

3. O que a função fibo deve retornar em (1)?

   Indicado no código abaixo.

```
Program Fibonacci;
Type
    ArrayType = Array of longint;
var
    fibs: ArrayType = nil;
    fibo_ncalls: longint = 0;
    fib2_ncalls: longint = 0;

function fibo ( n: Integer ): longint;
var
    i: integer;
begin
    fibo_ncalls += 1;
    if n <= 1 then begin
       fibo := n;
       exit;
    end;

    if (fibs = nil) or (length(fibs) <= n) then begin
        fibs := nil;
        SetLength(fibs, n+1);
        for i := 0 to n do
            fibs[i] := 0;
    end;

    // if fibs[*] was already calculated, do nothing
    if (fibs[n-1] = 0) then fibs[n-1] := fibo(n-1);
    if (fibs[n-2] = 0) then fibs[n-2] := fibo(n-2);

    fibs[n] := fibs[n-2]+fibs[n-1];

    fibo := fibs[n];
end;
```

4 Quarta questão (2 pontos)

Escreva as duas linhas que estão faltando na funçao **recursiva** fib2, que também executa apenas n chamadas recursivas, para que ela retorne o n-ésimo elemento da sequência de Fibonacci.
Nota: a primeira chamada da função fib2 é através da função fib, abaixo.

```
function fib2(n, p0, p1: longint): longint;
begin
    fib2_ncalls += 1;

    if n = 1 then
       fib2 := p1
    else
       fib2 := fib2(n - 1, p1, p0 + p1)
end;

function fib(n: longint): longint;
begin
    fib2_ncalls := 0;
    if n = 0 then
       fib := 0
    else
       fib := fib2(n, 0, 1)
end;
```