

USP - EACH
Organização de Computadores Digitais
1º Semestre - 2017

MIPS

Docente - Profa. Dra. Gisele da Silva Craveiro

Problema

3. Uma sequência de n números inteiros não nulos é dita *piramidal m -alternante* se é constituída por m segmentos: o primeiro com um elemento, o segundo com dois elementos e assim por diante até o m -ésimo, com m elementos. Além disso, os elementos de um mesmo segmento devem ser todos pares ou todos ímpares e para cada segmento, se seus elementos forem todos pares (ímpares), os elementos do segmento seguinte devem ser todos ímpares (pares).

Por exemplo, a sequência com $n = 10$ elementos:

12 3 7 2 10 4 5 13 5 11 é piramidal 4-alternante.

A sequência com $n = 3$ elementos:

7 10 2 é piramidal 2-alternante.

A sequência com $n = 8$ elementos:

1 12 4 3 13 5 12 6 não é piramidal alternante pois o último segmento não tem tamanho 4.

(a) Escreva uma função *bloco* que recebe como parâmetro um inteiro n e lê n inteiros do teclado, devolvendo um dos seguintes valores:

- 0, se os n números lidos forem pares;
- 1, se os n números lidos forem ímpares;
- 1, se entre os n números lidos há números com paridades diferentes.

(b) usando a função do item anterior, escreva um programa que, dados um inteiro $n \geq 1$ e uma sequência de n números inteiros, verifica se ela é piramidal m -alternante. O programa deve imprimir o valor de m ou dar a resposta *não*.

Solução Alto Nível

```
1 import java.util.Scanner;
2
3 public class EpOCD {
4
5     private static int bloco(int n) {
6
7         Scanner s = new Scanner(System.in);
8         int total = 2;
9
10        while (n > 0) {
11            n--;
12            int next = s.nextInt();
13
14            if (total == 2)
15                total = next % 2;
16
17            else if (total == 1) {
18
19                if (next % 2 != 0) {
20
21                } else {
22                    total = -1;
23                }
24
25            } else if (total == 0) {
26                if (next % 2 == 0) {
27
28                } else {
29                    total = -1;
```

```
35
36
37        return total;
38
39    }
40
41    public static boolean verificaPiramidavel(int elementos) {
42
43        int i = elementos;
44        for (int j = 0; j < elementos; j++) {
45            i -= j + 1;
46
47            if (i == 0) return true;
48        }
49
50        return false;
51
52    }
53
54    public static void main(String[] args) {
55
56        Scanner s = new Scanner(System.in);
57
58        int blocosALer = s.nextInt();
59
60        if (!verificaPiramidavel(blocosALer)){
61            System.out.println("não");
62            return;
63        }
```

Solução Alto Nível

```
67 int proxBlocoDeveSer = 0;
68 int j = 0;
69
70 for (int i = 0; i < blocosALer;) {
71
72     int numerosNoBloco = j + 1;
73     j++;
74
75
76     int blocoTotal = bloco(numerosNoBloco);
77
78     if (blocoTotal == -1){
79         System.out.println("não");
80         return;
81     }
82
83
84     if (i == 0) {
85         proxBlocoDeveSer = inverteBit(blocoTotal);
86
87     }else {
88         if (blocoTotal != proxBlocoDeveSer) {
89             System.out.println("não");
90             return;
91         }
92
93         proxBlocoDeveSer = inverteBit(blocoTotal);
94     }
95 }
```

```
94     }
95
96     i = numerosNoBloco + i;
97
98 }
99
100 System.out.println(j);
101
```

- 50 linhas de código
- Número “ilimitado” de variáveis
- Funções simples
- Muitas ferramentas

Solução Assembly

- Não cabe nos slides – 116 linhas de código
- Funções simples devem ser escaladas para funções mais complexas
- Número limitado de variáveis temporárias (registradores)
- Menos “legível” – ruim de fazer manutenção
- Depende muito da arquitetura

OBRIGADO!!

```
System.out.println("Java Rainha Assembly Nadinha");
```