

1.4 Estruturas de Dados

1.4.2. exercício extra

Escrever um algoritmo que leia as notas da primeira prova de Pacotes (notas inteiras, de 0 a 30, o aluno “*Ninguém*”, com nota igual a -1 é o sinal de final de arquivo), monte a respectiva tabela de frequências e imprima o resultado.

Em outras palavras, o algoritmo deverá ler o seguinte banco de dados:

Fulano	13
Sicrano	18
Beltrano	13
...	
Ninguém	-01

Deve, em seguida, criar e imprimir a seguinte tabela (note que ficam de fora as notas de frequência nula):

Nota	Frequência
...	...
13	2
18	1
...	...

1.4 Estruturas de Dados

1.4.2. exercício extra

Sugestão:

Se você estiver completamente perdido, sugiro começar olhando o Exemplo 4, Seção 1.4.1, da reserva de passagens aéreas. Antes de começar a escrever o algoritmo, também é bom definir as estruturas de dados necessárias, que poderão ser tão simples quanto as que se seguem:

	Notas	Frequência	NomeAluno	NotaAluno
1	0	4	Fulano	30
2	1	2		
3	2	0		
⋮		⋮		
31	30	10		

1.4 Estruturas de Dados

1.4.2. exercício extra

Solução:

A seguir, apresentamos uma solução possível para o problema. Inicialmente, desenvolvemos uma *primeira versão* do algoritmo, já identificando a estrutura de repetição que deverá estar presente, mas deixando para mais tarde alguns detalhes (isto é, os refinamentos):

```
algoritmo
  defina os tipos das variáveis
  atribua valores iniciais necessários
    { leia e processe entrada }
  leia nomeAluno, notaAluno
  enquanto notaAluno <> -1 faça
    processe nota
    leia nomeAluno, notaAluno
  fim enquanto
  escreva tabela de frequências
fim algoritmo
```

1.4 Estruturas de Dados

1.4.2. exercício extra

Solução (cont.):

Refinando-se o processamento da nota lida e sua contribuição para a frequência, tem-se:

```
ref.: processe nota
        { encontre a posição da nota }
        i ← notaAluno+1
        { atualize a frequência }
        frequencia[i] ← frequencia[i]+1
fim ref.
```

Refinando-se a atribuição de valores iniciais necessários, tem-se:

```
ref.: atribua valores iniciais necessários
        para i de 1 até 31 faça
            notas[i] ← i-1
            frequencia[i] ← 0
        fim para
fim ref.
```

1.4 Estruturas de Dados

1.4.2. exercício extra

Solução (cont.):

Refinando-se a escrita da tabela de frequências:

```
ref.: escreva tabela de frequências
      escreva “nota”, “frequência”
      para i de 1 até 21 faça
          se (frequencia[i]=0) então
              { não faça nada }
          senão
              escreva notas[i], frequencia[i]
          fim se
      fim para
fim ref.
```

Refinando-se a declaração de variáveis:

```
ref.: defina os tipos das variáveis
      declare notas[1:31], frequencia[1:31], notaAluno numérico
      declare nomeAluno literal
fim ref.
```

1.4 Estruturas de Dados

1.4.2. exercício extra

Solução (cont.):

Inserindo-se os refinamentos nos seus respectivos lugares, temos o algoritmo completo:

```
algoritmo
    { defina os tipos das variáveis }
    declare notas[1:21], frequencia[1:21], notaAluno numérico
    declare nomeAluno literal
    { atribua valores iniciais necessários }
    para i de 1 até 21 faça
        notas[i] <- i-1
        frequencia[i] <- 0
    fim para
    { leia e processe entrada }
    leia nomeAluno, notaAluno
    enquanto notaAluno <> -1 faça
        { processe notas }
        { encontre a posição da nota }
        i <- notaAluno+1
        { atualize a frequência }
        frequencia[i] <- frequencia[i]+1
        { leia dados do próximo aluno }
        leia nomeAluno, notaAluno
    fim enquanto
    { escreva tabela de frequências }
    escreva "nota", "frequência"
    para i de 1 até 21 faça
        se (frequencia[i]=0) então
            { não faça nada }
        senão
            escreva notas[i], frequencia[i]
        fim se
    fim para
fim algoritmo
```