

EP3 - Linguagem C - Compactação LZW

Critérios de Correção

1) A avaliação automática do VPL/e-Disciplinas determinou a nota de base no exercício. A correção automática conferiu o resultado de 8 testes realizados na função *main()* do arquivo *testa_compactacao.c* (cuja implementação completa foi fornecida aos alunos). Cada teste vale 1,25 pontos, totalizando 10 pontos. A entrada fornecida ao programa e a saída esperada de cada teste estão listados no final deste documento.

Caso o código entregue pelo grupo tenha falhado nos testes do VPL devido a erros simples de sintaxe ou lógica, esses erros foram corrigidos e o código foi testado manualmente em seguida, para a obtenção da nota de base. Da nota obtida, descontou-se 2,0 pontos (como penalização pelos erros).

2) Sobre a nota obtida depois do passo 1, aplicou-se os seguintes descontos de nota quando pertinente:

	Descrição do Problema	Desconto
	Mudou a definição da estrutura <i>tDicionario</i> e/ou os nomes ou parâmetros das funções em <i>dicionario.h</i> ou em <i>compactacao.h</i>	5,0
	Não usou as funções declaradas em <i>dicionario.h</i> na implementação das funções declaradas em <i>compactacao.h</i>	5,0
	Gerou de forma artificial a resposta esperada para um teste	2,0 por teste
	Código pouco legível (por exemplo, mal indentando, com nomes de variáveis ruins, com falta de espaços entre as palavras e demais símbolos, etc.)	2,0
	Não assinou o cabeçalho do EP nos arquivos <i>dicionario.c</i> e <i>compactacao.c</i>	0,5

Para outros tipos de problemas, o desconto da nota foi definido de acordo com a gravidade do caso.

Testes Executados pelo Avaliador Automático

1) Criação do dicionário de compactação do exemplo 1

```
Case = Dicionário do exemplo1.txt
input = 1
exemplo1.txt
output = Digite modo do programa:
Digite o nome do arquivo de texto a ser compactado:
>>> IMPRESSÃO DO DICIONÁRIO DE COMPACTAÇÃO | Arquivo: exemplo1.txt
 128 | as
 129 | s_
 130 | _a
 131 | asa
 132 | as_
 133 | _d
 134 | da
 135 | as_c
 136 | ca
 137 | asas
```

2) Compactação do exemplo 1

```
Case = Compactação do exemplo1.txt
input = 2
exemplo1.txt
exemplo1_compactado.bin
output = Digite modo do programa:
Digite o nome do arquivo de texto a ser compactado:
Digite o nome do arquivo compactado gabarito:
>>> TESTE DE COMPACTAÇÃO | Arquivo: exemplo1.txt
Arquivo de texto de entrada: exemplo1.txt - Tamanho: 17 bytes
Arquivo binário com o texto compactado: compactado.bin - Tamanho: 22 bytes
Tamanho esperado para o arquivo compactado: 22 bytes
O conteúdo do arquivo compactado está CORRETO.
```

3) Criação do dicionário de compactação do exemplo 1

```
Case = Dicionário do exemplo1_compactado.bin
input = 3
exemplo1_compactado.bin
output = Digite modo do programa:
Digite o nome do arquivo binário a ser descompactado:
>>> IMPRESSÃO DO DICIONÁRIO DE DESCOMPACTAÇÃO | Arquivo:
exemplo1_compactado.bin
 128 | as
 129 | s_
 130 | _a
 131 | asa
 132 | as_
 133 | _d
 134 | da
 135 | as_c
 136 | ca
```

4) Descompactação do exemplo 1

```
Case = Descompactação do exemplo1.txt
input = 4
exemplo1_compactado.bin
exemplo1.txt
output = Digite modo do programa:
Digite o nome do arquivo binário a ser descompactado:
Digite o nome do arquivo descompactado gabarito:
>>> TESTE DE DESCOMPACTAÇÃO | Arquivo: exemplo1_compactado.bin
Arquivo binário com os dados compactados: exemplo1_compactado.bin - Tamanho:
22 bytes
Arquivo com o texto descompactado: descompactado.txt - Tamanho: 17 bytes
Tamanho esperado para o arquivo descompactado: 17 bytes
O conteúdo do arquivo descompactado está CORRETO.
```

5) Compactação do exemplo 2

```
Case = Compactação do exemplo2.txt
input = 2
exemplo2.txt
exemplo2_compactado.bin
output = Digite modo do programa:
Digite o nome do arquivo de texto a ser compactado:
Digite o nome do arquivo compactado gabarito:
>>> TESTE DE COMPACTAÇÃO | Arquivo: exemplo2.txt
Arquivo de texto de entrada: exemplo2.txt - Tamanho: 18323 bytes
Arquivo binário com o texto compactado: compactado.bin - Tamanho: 12086 bytes
Tamanho esperado para o arquivo compactado: 12086 bytes
O conteúdo do arquivo compactado está CORRETO.
```

6) Descompactação do exemplo 2

```
Case = Descompactação do exemplo2.txt
input = 4
exemplo2_compactado.bin
exemplo2.txt
output = Digite modo do programa:
Digite o nome do arquivo binário a ser descompactado:
Digite o nome do arquivo descompactado gabarito:
>>> TESTE DE DESCOMPACTAÇÃO | Arquivo: exemplo2_compactado.bin
Arquivo binário com os dados compactados: exemplo2_compactado.bin - Tamanho:
12086 bytes
Arquivo com o texto descompactado: descompactado.txt - Tamanho: 18323 bytes
Tamanho esperado para o arquivo descompactado: 18323 bytes
O conteúdo do arquivo descompactado está CORRETO.
```

7) Compactação do exemplo 3

```
Case = Compactação do exemplo3.txt
input = 2
exemplo3.txt
exemplo3_compactado.bin
output = Digite modo do programa:
```

Digite o nome do arquivo de texto a ser compactado:
Digite o nome do arquivo compactado gabarito:
>>> TESTE DE COMPACTAÇÃO | Arquivo: exemplo3.txt
Arquivo de texto de entrada: exemplo3.txt - Tamanho: 47368 bytes
Arquivo binário com o texto compactado: compactado.bin - Tamanho: 25310 bytes
Tamanho esperado para o arquivo compactado: 25310 bytes
O conteúdo do arquivo compactado está CORRETO.

8) Descompactação do exemplo 3

Case = Descompactação do exemplo3.txt
input = 4
exemplo3_compactado.bin
exemplo3.txt
output = Digite modo do programa:
Digite o nome do arquivo binário a ser descompactado:
Digite o nome do arquivo descompactado gabarito:
>>> TESTE DE DESCOMPACTAÇÃO | Arquivo: exemplo3_compactado.bin
Arquivo binário com os dados compactados: exemplo3_compactado.bin - Tamanho:
25310 bytes
Arquivo com o texto descompactado: descompactado.txt - Tamanho: 47368 bytes
Tamanho esperado para o arquivo descompactado: 47368 bytes
O conteúdo do arquivo descompactado está CORRETO.