

BCC202 – Estruturas de Dados I (2018-02)

Departamento de Computação - Universidade Federal de Ouro Preto - MG

Trabalho Prático 01 (TP01)

Data de entrega: 01/10/2018 até 22:00.

Atenção: *O que vale é o horário do RunCodes, e não do seu, ou do meu, relógio.*

Procedimento para a entrega:

1. Para cada questão, implemente sua solução.
 2. Para testar suas soluções, implemente um único método *main()*, que poderá conter, por exemplo, um *Menu* de interações e possibilidades do usuário especificar os dados de entrada.
 3. Especifique o *Makefile* com as instruções necessárias para compilação e execução do seu código, sendo que o *Makefile* deve conter também o redirecionamento da entrada e da saída (e.g., `./prog.exe < input.txt > output.txt`.)
 4. Compacte em um único diretório o seu código fonte juntamente com o *Makefile*, o arquivo de entrada e o arquivo de saída usados para testes (e.g., *input.txt*, *output.txt*).
 5. Faça a entrega do arquivo compactado, obrigatoriamente em formato *.zip*, no *RunCodes*, na tarefa correspondente.
- Não utilize caracteres acentuados ou especiais para nomes de pastas, arquivos e na especificação de comentários no código.
 - Implemente em conformidade com boas práticas para reuso e modularização do código.

- Bom trabalho!

Questão 01 (4 pontos)

Considere um sistema de matrícula que deve ser composto pelos seguintes Tipos Abstratos de Dados (TADs): *aluno*, representado por seu nome, CPF, número de matrícula, curso e data de ingresso; *disciplina*, representada por seu nome, código, carga-horária (4 ou 8 horas), curso e sala em que é oferecida; *professor*, representado por seu CPF, nome e departamento onde está lotado; *atestado de matrícula*, composto por um aluno e um conjunto de disciplinas, não ultrapassando 28 horas-aulas por semana para cada discente; e *encargo didático*, composto um professor e um conjunto de disciplinas, não ultrapassando 16 horas semanais de encargo para cada docente. Especifique e implemente, utilizando conceitos de TADs e alocação dinâmica de memória, uma versão simples desse sistema de matrícula, em que seja possível, via *menu principal*, criar/remover/atualizar/buscar alunos, disciplinas, professores, atestados de matrícula e encargos didáticos.

Questão 02 (4 pontos)

Dado um mapa na forma de uma matriz de caracteres, implemente uma função recursiva que encontre e imprima um caminho entre a posição [0,0] e a posição [9,X], $0 \leq X \leq 9$. Um exemplo de mapa está mostrado a seguir, com saída em [9,9]. No mapa, os asteriscos (***) representam paredes, o canto *inicio* representa o ponto inicial, o canto *fim* o ponto final e (000) representa um caminho disponível. Deverá ser possível entrar com a matriz que representa o labirinto, tendo ao menos um caminho possível de [0,0] até [9,X]. Especifique a equação de recorrência do seu algoritmo.

Labirinto:

```
0 inicio *** 000 000 000 000 000 000 000 000
1 000 000 *** 000 *** ** * ** * ** * **
2 *** 000 000 *** 000 000 000 *** 000 ***
3 *** ** * 000 000 *** 000 000 *** 000 000
4 000 *** ** * 000 000 *** 000 *** 000 000
```

```

5 000 000 000 *** 000 *** 000 *** 000 000
6 000 000 000 000 000 *** 000 *** 000 000
7 000 *** *** *** *** *** 000 *** *** ***
8 000 000 000 000 000 000 000 000 000
9 *** *** *** *** *** *** *** *** *** fim

```

Legenda:

000 = caminho livre

*** = parede

Questão 03 (2 pontos)

Sobre a exponenciação, considere as informações abaixo:

- Se n é par:

$$x^n = [(x^2)^{(n/2)}] \quad (1)$$

- Se n é ímpar:

$$x^n = x * [(x^2)^{(n-1)/2}] \quad (2)$$

- (1.0) Com base nas afirmações anteriores, implemente uma função recursiva para o cálculo da exponenciação.
- (1.0) Com base na implementação anterior, formule, expanda e resolva a equação de recorrência do seu algoritmo. NÃO é preciso utilizar o Princípio de Indução Matemática para provar a corretude da complexidade encontrada.