

UFRJ-NCE \_ Professora Priscila \_ 2018/1

Disciplina: COS110 – Algoritmos e Programação

Curso: Engenharia de Controle e Automação

### Trabalho em Grupo

- O trabalho deve ser executado por um grupo de , no máximo, 3 componentes.
- A entrega deve incluir num arquivo compactado, os arquivos fonte (.c), objetos (.o) e executáveis (.exe) .
- O grupo inteiro deve participar da apresentação/argüição sobre o trabalho.
- Abaixo, seguem 3 sugestões de temas para o básico do trabalho, valendo 7,0 de 10,0.
- Os outros 3,0 pontos advirão de funcionalidades extra incorporadas pelo grupo ao trabalho.
- Caso o grupo deseje propor outro tema, deve fazê-lo até sexta-feira 22/06.

#### 1. Álbum de figurinhas:

- i. Inspiração: álbum de figurinhas da COPA 2018. Deve-se poder verificar estado atual do álbum.
- ii. Composto por páginas, cada uma com uma quantidade específica de figurinhas .
- iii. Figurinhas identificadas pelo seu número.
- iv. Cada página tem uma disposição das suas figurinhas.
- v. Simular “compra” de pacotinhos de figurinhas.
- vi. Administrar duplicatas e trocas de figurinhas com outros usuários.

#### 2. Bolão da COPA 2018 (ou outro):

- i. Inspiração: bolão de resultados da COPA 2018. Deve-se poder verificar estado atual do bolão.
- ii. Deve ser possível cadastrar/excluir participantes.
- iii. Cada participante tem identificação e dados para contato, além de saldo no bolão.

- iv. A cada jogo, cada participante deve informar quanto vai apostar e qual o resultado da aposta. Aposta só no resultado: Vitória de A/Vitória de B/ Empate OU Aposta exata nos gols de A e de B. Também pode ser possível informar quantia fixa para todos os jogos e somente o resultado jogo a jogo.
- v. A partir do resultado de um jogo, calcular como ficam os saldos dos participantes. Ao final do campeonato (COPA 2018), listar o saldo final de cada participante.

### 3. Alocação Dinâmica de Memória Personalizada:

- i. Escreva um programa que receba do usuário quantidades de bytes a serem alocadas/liberadas, simulando um gerenciador de *heap*.
- ii. A alocação deve ser feita por quatro funções construídas pelo grupo: *myalloc*, *myfree*,
- iii. *mycalloc* e *myrealloc*
- iv. O comportamento do “*myheap*” supõe que cada região disponível (bloco de tamanho variável) contém duas informações: (i) tamanho do bloco e (ii) ponteiro para próximo bloco, conforme exemplificado pela figura abaixo.
- v. Inicialmente o “*myheap*” está todo livre e é composto por um único bloco de tamanho máximo (arbitre um valor visualizável). A cada interação com o usuário, o estado do “*myheap*” deve ser exibido.

