

Curso: Engenharia de Software
Disciplina: Estruturas de Dados e Algoritmos
Carga Horária: 60 horas
Professor: Matheus de Sousa Faria

Semestre/Ano: 02/2017
Créditos: 04
Turma: C

Laboratório 01

Memória

1. Descrição Geral

Este laboratório tem como objetivo avaliar o aprendizado através da prática do conteúdo, assim como por questões teóricas.

Todas as questões da seção de questões serão avaliadas, se, e somente se, cumprirem as normas abaixo:

1. As respostas das questões práticas devem ser feitas nas linguagens C ou C++.
2. Os arquivos de código devem ser nomeado com sua matrícula sem a "/" (barra), underline (_), número ou letra (maiúscula) da questão.
Exemplo: 1600342312_1.c ou 15012434_A.c
3. Todas as respostas devem ser originais e de autoria do próprio aluno, qualquer cópia ou plágio irá desqualificar TODAS as questões para correção.
4. As respostas devem ser enviadas via moodle na data indicada na plataforma, não serão aceitos envios tardios.
5. Todas as questões teóricas devem ser respondidas em um único arquivo, que deve ser nomeado com sua matrícula sem "/" (barra), underline (_), a palavra "teoricas" (sem acento e tudo minúsculo). E a extensão deve ser ".txt".
Exemplo: 1600342312_teoricas.txt
6. O formato do arquivo das questões teóricas deve ser o seguinte:
 - a. Para cada questão teórica deve-se colocar o número ou letra (Maiúscula) da questão, seguido por ":" (dois pontos). Na linha seguinte a identificação, deve haver a resposta da questão.
 - b. Salte uma linha entre duas questões
7. Questões teóricas de múltipla escolha não devem conter nada além da opção em sua resposta
8. Em todas as questões práticas, pontos serão descontado por: má indentação de código, maus nomes de variáveis, e falta de comentários.

Todas as respostas devem ser colocadas em uma pasta com o nome igual a sua matrícula, compactadas em formato ZIP, e enviadas para o moodle.

Um exemplo de formatação pode ser encontrado no moodle da disciplina.

2. Questões

Questão 1 (5 pontos)

Implemente a estrutura de dados vector, como foi discutido em sala de aula. O tipo interno da sua estrutura deve ser inteiro (int). As assinaturas das funções deve ser essas:

```
Vector create(int capacity);  
void destroy(Vector * vector);  
void push_back(Vector * vector, int element);  
void pop_back(Vector * vector);  
void erase(Vector * vector, int i);  
void clear(Vector * vector);  
int back(Vector * vector);  
int front(Vector * vector);  
int at(Vector * vector, int i);  
unsigned int size(Vector * vector);  
unsigned int capacity(Vector * vector);
```

O que as funções devem fazer está explicado no PDF da aula da semana. E juntamente com este documento se encontra um arquivo “vector.c”, este arquivo possui uma função main que deve ser utilizada para testar o seu programa.

Caso seu programa falhe ao compilar ou ao executar em algum caso de teste, 1 ponto será descontado para cada caso que falhar. Com a nota mínima de 0.

Questão 2 (2 pontos)

Implemente, utilizando a alocação dinâmica, uma matriz de inteiros de tamanho variável de linhas e colunas. Faça um programa que aceite duas matrizes e retorne a soma das duas.

Questão 3 (1 ponto)

Dado o código abaixo, indique quantas variáveis serão colocadas no segmento de dados inicializados, bss, heap e stack. Sua resposta deve ser no formato:

```
data: 4  
bss: 2
```



heap: 7
stack: 30

Neste exemplo, 4 variáveis pertencem ao segmento data, 2 ao bss, 7 a heap, e 30 a stack.

Abaixo o código que você deve analisar:

```
int function(int a) {  
    int b = 32;  
    b = a + b;  
  
    return b;  
}  
  
int functionA(double * a) {  
    int * b = (int *) malloc(3 * sizeof(int));  
    static double c = 56;  
  
    return 0;  
}  
  
int C = 45, D, E;  
  
int main() {  
    static char letra = 'A', letra1;  
    int b = 3, f;  
  
    return 0;  
}
```

Questão 4 (1 ponto)

Considerando que:

```
int a = 32;  
int b[] = {1, 2, 3};
```

Identifique os tipos das seguintes expressões:

1. a;
2. &a;



3. `&&a;`
4. `*&a;`
5. `&b;`
6. `b;`
7. `b[0];`
8. `&(b[0]);`
9. `b[1];`
10. `&*&a;`

A resposta dessa questão deve seguir o seguinte formato

```
1: char
2: double ****
3: float *
```

O número da expressão, seguido por “:”, e o tipo da linguagem C para aquela expressão. **Não** coloque o nome completo do tipo, por exemplo, Inteiro.

Questão 5 (1 ponto)

Quais são as vantagens e desvantagens da alocação dinâmica, exemplifique sua resposta mostrando como se aloca e desaloca um array de structs.