

Trabalho Prático – Valor 30 pontos

Objetivo do Projeto:

O objetivo deste trabalho é criar uma aplicação usando a linguagem de programação C (padrão ANSI/ISO) para ilustrar o uso de uma das estruturas de dados: lista simplesmente encadeada, lista simplesmente encadeada circular, lista duplamente encadeada, lista duplamente encadeada circular, fila, pilha ou deque em uma situação real. Lembrando que a estrutura de dados deverá ser útil para o cenário de aplicação escolhido, tornando mais simples a solução de um problema. Incluindo o código fonte do sistema, deverá ser apresentada uma proposta descrevendo o cenário de aplicação com o problema a ser solucionado.

Número de participantes: grupos de até 3 (três) integrantes.

→ **1ª Etapa parte 1 – Proposta:** Um parágrafo com a descrição em linhas gerais do projeto a ser desenvolvido, e outro parágrafo justificando o uso da estrutura de dados (máximo: 120 palavras);

→ **1ª Etapa parte 2 – Enunciado:**

1. *Características básicas.*

- ✓ Serão consideradas situações reais:
 - 1) Situações relacionadas ao dia a dia do aluno, ou
 - 2) História da Universidade Federal de Viçosa, campus de Rio Paranaíba, ou
 - 3) Riquezas e cultura da região do Alto Paranaíba, ou
 - 4) Situações relacionadas à área de trabalho do seu curso.
- ✓ O texto com o enunciado do problema deve conter a descrição detalhada do projeto, com preferencialmente 3 (três) parágrafos, contendo:
 - 1) Uma introdução explicando o problema de forma geral;
 - 2) Um parágrafo descrevendo o cenário da aplicação;
 - 3) Um (ou mais) parágrafo(s) com os dados para solucionar o problema.
- ✓ O texto deve conter no máximo **500 palavras**.
- ✓ Não serão aceitos enunciados que foram trabalhados em sala de aula, roteiros de aula prática ou que estejam nos slides.

→ **2ª Etapa – Codificação:**

1. *Características básicas.*

- 1) Ser implementado em linguagem C (com extensão de arquivo “.c”):
 - a. **Nota:** Arquivos com extensão “.cpp” são da linguagem C++ e, portanto, não serão aceitos.
- 2) Ter um menu para escolha da operação a ser executada, seja de execução de entrada de dados ou algum processamento sobre os dados.
- 3) Implementar a solução obedecendo a estrutura de dados sorteada em sala de aula (lista simplesmente encadeada, lista duplamente encadeada, fila, pilha, deque, etc):
 - a. **Nota:** Trabalhos implementados com uma estrutura que não seja a mesma sorteada serão avaliados com o máximo de 50% da avaliação da codificação.
- 4) A estrutura de dados deve ser codificada utilizando a implementação dinâmica.
 - a. **Nota:** Não serão aceitos trabalhos que forem codificados usando a implementação estática da estrutura de dados.
- 5) Possuir pelo menos quatro operações para manipulação da estrutura de dados sorteada que sejam necessárias ao contexto da aplicação. Operações para inicializar, verificar estrutura vazia, alocar/desalocar não serão consideradas.
- 6) Os grupos que forem sorteados com a estrutura lista encadeada (simplesmente, duplamente, circular), além das quatro operações indicadas no passo 5, devem

também implementar uma operação para organizar a lista quando uma busca for realizada (Mover para Frente, Transposição ou Contagem).

- 7) Os grupos que forem sorteados com a estrutura pilha, fila ou deque além das quatro operações indicadas no passo 5, devem também implementar uma operação recursiva.

Obs.: No passo 6 e 7, lembre-se que o método escolhido deverá ser adequado ao contexto da aplicação.

2. Características extras (que serão usadas na avaliação):

1. Boas práticas de programação.
 - Comentários no código facilitando o entendimento (não é em todas linhas);
 - **indentação** do código (4 espaços em vez de tab);
 - Validação de entrada de dados;
 - etc.

```
#include <stdio.h>

void neoStart(){ // Neo vai para Matrix
printf("\nMatrix iniciando...\n");
}

int main(){
int valorNum,i;

printf("Digite um valor:");
scanf("%d",&valorNum);

for(i=0;i<valorNum;i++){
printf("\nOi mundo: %d",valorNum);
}

if(valorNum==9){
printf("\nBem-vindo a Matrix\n");
neoStart();
} else{
printf("\nFim do Programa\n");
}

return 0;
}
```

Prazos para entrega:

→ 1ª Etapa:

- Apresentação da proposta em sala de aula: o Data: **28 e 30/05/2019**.
 - Local/Horário: sala/horário da disciplina de Estrutura de Dados
 - Ordem de apresentação: será definida por sorteio
 - Tempo de apresentação: 10 a 15 min por grupo
- Prazo para entrega da proposta: **28/05/2019** (em sala de aula).
- Prazo para entrega do Enunciado via PVANet: **09/06/2019**.

→ 2ª Etapa:

- Prazo para entrega do projeto final (Enunciado + Codificação): **30/06/2019**
- Entrevista:
 - ⇒ Data: **02/07/2019 e 05/07/2019**
 - ⇒ Local/Horário: sala/horário da disciplina de Estrutura de Dados

- ⇒ Ordem de entrevista: será definida por sorteio
- ⇒ Tempo de entrevista: 10 a 15 min por grupo

Avaliação:

A avaliação do trabalho será feita de duas formas:

- 1) Avaliação do enunciado
 - 2) Avaliação do código fonte e entrevista
- A avaliação do enunciado poderá assumir até 5 pontos.
 - A avaliação do código fonte e entrevista poderá assumir até 25 pontos.

Entrega do trabalho:

→ Via PVANet:

- O projeto deverá ser entregue via PVANet em um único arquivo compactado/zipado (“.zip”), nomeado da seguinte forma **Matricula1_Matricula2_Matricula3.zip**, contendo:
 - Os códigos fontes do programa (arquivos “.c” e talvez “.h”), nomeados da seguinte forma:
 - **Matricula1_Matricula2_Matricula3.c**
 - O Enunciado elaborado contendo:
 - ✓ Capa (formatação ABNT) incluindo título do projeto e nome/matricula dos integrantes;
 - ✓ Enunciado com a descrição detalhada do projeto implementado;
 - ✓ Os dois itens anteriores em um único arquivo PDF, nomeado da seguinte forma:
 - **Matricula1_Matricula2_Matricula3.pdf**
- O arquivo deverá ser enviado por apenas um componente do grupo;

→ Observações:

- Todos os trabalhos em formato digital serão recebidos **somente** via PVANet (Opção “Entrega de Trabalhos”) dentro do prazo preestabelecido.
- Arquivos com problema são de inteira responsabilidade do aluno.
- O professor corrigirá o arquivo que for enviado.
- Se não for enviado o código fonte, o aluno automaticamente receberá a nota **zero**.
- Casos de flagrantes de cópias de trabalhos ocasionarão nota **zero** para todos os trabalhos semelhantes.
- Os grupos que não comparecerem a entrevista terão o trabalho computado com nota **zero**.
- Os códigos devem ser feitos em **linguagem C padrão ANSI/ISO**, ou seja, devem ter somente funções e operações que possam ser compiladas em qualquer compilador de qualquer sistema operacional.

➤ Para certificar que o código-fonte está correto dentro do padrão ANSI/ISO, teste em compiladores do Windows e do GNU/Linux.

➤ Comando:

gcc -Wall -Wextra -Wpedantic -O3 -ansi meuProg.c -std=c99 -o meuProg.out

O std pode ser c99 ou c11, para c17 e c18 vai precisar do GCC atual. Leia mais sobre:

➤ <https://fig.if.usp.br/~esdobay/c/gcc.html>

➤ https://linux.ime.usp.br/~lucasmmg/livecd/documentacao/documentos/terminal/Compilando_um_arquivo_em_C.html

- Um ponto extra no projeto para a equipe que utilizar o Git (enviar o repositório junto com o trabalho).