

Primeiro Trabalho Computacional – Valor 2,0

Objetivo do Trabalho:

- Implementar (na Linguagem C) e entregar um dos exercícios abaixo (que será sorteado):

1) Jogo da Velha (2 jogadores)

Obs.: Todas as jogadas têm que ser armazenadas no arquivo **Velha.txt**.

2) Jogo da Forca (2 jogadores)

O jogo começa com o jogador 1 digitando uma palavra oculta (de, no máximo, 10 letras), que é representada na tela por um conjunto de traços que servem para indicar as letras da palavra digitada. O jogador 2 deve dar palpites de letras que acreditam estar presentes na palavra oculta. A cada acerto, a letra digitada é colocada na posição em que aparece na palavra. Quando ocorre um erro é desenhado uma parte do homem-palito. O jogo termina quando a palavra for descoberta ou quando o homem-palito estiver totalmente desenhado (palavra não descoberta). Em todos estes dois casos, a palavra oculta deve ser apresentada.

Obs.: Todas as etapas do jogo têm que ser armazenadas no arquivo **Forca.txt**.

3) Jogo da Memória (1 jogador)

O computador deve apresentar na tela 7 números distintos (gerados aleatoriamente de 1 até 50) durante 5 segundos. O jogador deve tentar digitar tais números na ordem em que os mesmos aparecem.

Obs.: Todas as etapas do jogo têm que ser armazenadas no arquivo **Memoria.txt**.

4) Batalha Naval (2 jogadores)

Cada jogador tem um campo de tamanho 6 x 6, sendo constituído, portanto, de 36 blocos. Em cada campo há 4 navios escondidos (um em cada bloco), que são determinados aleatoriamente. O objetivo de cada jogador é descobrir onde estão os 4 navios do outro jogador e acertá-los. A cada tiro dado (indicação de uma linha e uma coluna) é dito se o jogador acertou algum navio. O jogo só acaba quando um jogador descobrir e afundar todos os navios do outro jogador.

Obs.: Todas as etapas do jogo têm que ser armazenadas no arquivo **Batalha.txt**.

5) Agenda

Faça um algoritmo para gerenciar uma agenda de contatos. Para cada contato armazene o **nome** (sem espaços), o **telefone** e o **aniversário** (dia e mês). O algoritmo deve permitir:

- (1) Inserir um contato;
- (2) Remover um contato pelo nome;
- (3) Pesquisar um contato pelo nome, apresentando as informações na tela;
- (4) Listar na tela todos os contatos;
- (5) Imprimir na tela os aniversariantes do mês;
- (6) Sair da execução do algoritmo.

A agenda tem que ser armazenada em um arquivo texto chamado “**Agenda.txt**”, sobre o qual as operações de (1) até (5) devem ser realizadas. Obviamente as opções de (2) até (5) só podem ocorrer se houver algum contato inserido no arquivo. Utilize um **procedimento** para cada operação de (1) até (5). Utilize um **menu** para apresentar as opções.

Exemplo de um arquivo texto:

Agenda.txt

BrunaSouza 999109785 10 05
CarlosSilva 27336598 04 06
IgorPereira 27326485 25 07
LarissaCastro 999845731 18 09
MarcosMedeiros 998486487 20 04

6) Vendas

Faça um algoritmo para gerenciar registros de vendas dos funcionários de uma loja. Para cada venda armazene o **código da venda** (gerado aleatoriamente de 1 até 1000), o **nome do vendedor** (sem espaços), o **valor da venda**, além do **dia** e do **mês da venda**. O algoritmo deve permitir:

- (1) Inserir uma venda (o código da venda não pode ser repetido);
- (2) Remover uma venda pelo código da venda;
- (3) Listar na tela todas as vendas de um vendedor pesquisando pelo nome do vendedor;
- (4) Listar na tela todas as vendas de um vendedor em um determinado dia ou mês;
- (5) Imprimir na tela o valor total de vendas de um vendedor em um determinado dia ou mês;
- (6) Sair da execução do algoritmo.

Os registros de vendas têm que ser armazenados em um arquivo texto chamado “**Vendas.txt**”, sobre o qual as operações de (1) até (5) devem ser realizadas. Obviamente as opções de (2) até (5) só podem ocorrer se houver alguma venda inserida no arquivo. Utilize um **procedimento** para cada operação de (1) até (5). Utilize um **menu** para apresentar as opções.

Exemplo de um arquivo texto:

Vendas.txt

12	BrunoSouza	785.54	10	05
147	CarlosSilva	415.89	04	06
48	IgorPereira	1485.12	25	07
245	LarissaCastro	658.79	18	09
675	MarcosMedeiros	59.60	20	04

7) Conta Bancária

Faça um algoritmo para gerenciar a conta bancária de um cliente. O algoritmo deve permitir:

- (1) Depositar;
- (2) Sacar;
- (3) Transferir;
- (4) Ver Saldo na tela;
- (5) Ver Extrato na tela;
- (6) Sair da execução do algoritmo.

As operações de (1) até (3) devem ser registradas em um arquivo texto chamado “**Movimentacoes.txt**”. Quando a opção (5) for selecionada, o algoritmo deve ler o arquivo e imprimir todas as informações contidas nele. As opções de (2) até (5) só podem ocorrer se algum depósito já tiver sido realizado. Utilize um **procedimento** para cada operação de (1) até (5). Utilize um **menu** para apresentar as opções.

Exemplo de um arquivo texto:

Movimentacoes.txt

Deposito 1000.00
Saque 300.00
Deposito 100.00
Transferencia 150.00
Deposito 50.00

Importante

- O trabalho deve ser feito em grupo de, **no máximo**, 5 pessoas.
- Procedimentos e/ou funções têm que ser utilizados.
- Não é permitido declarar variáveis globais, exceto o #define.
- Todos os vetores e matrizes utilizados têm que ser declarados na função main.
- O grupo deve enviar o código (arquivo .c) até às **23h59** do dia **24/11/2019** (domingo) via e-mail para:

philippeal@yahoo.com.br
- Para os grupos que ficaram com os jogos, uma descrição de como o jogo deve ser jogado (arquivo .pdf) também deve ser enviado junto com o código.
- Apenas um e-mail por grupo deve ser enviado. Isto é, não há a necessidade de cada componente do grupo enviar. É recomendável enviar uma cópia do e-mail para os demais componentes do grupo.
- Trabalhos que não estão compilando não serão aceitos. Portanto, não é necessário enviar, receberão nota **ZERO**.
- Cuidado com trabalhos iguais/parecidos/com a mesma ideia de resolução. Receberão nota **ZERO**. Alterar os nomes das variáveis não torna os trabalhos diferentes.
- Caso o grupo esteja com dúvidas, tire-as com o professor **EM SALA, NÃO POR E-MAIL**, ou com os monitores. **NÃO DEIXE PARA A ÚLTIMA HORA**.

- O **ASSUNTO** do e-mail deve ter a seguinte formatação:

PE-T1-PrimeiroNomeDosAlunos

Exemplo de e-mail do grupo (fictício) formado pelos alunos Lucas Pereira, Renato da Silva e Miguel dos Santos que fizeram o Jogo da Força:

Para: philippeal@yahoo.com.br

De: Lucas Pereira

Assunto: PE-T1-Lucas-Renato-Miguel-JogodaForca

Anexo: PE-T1-Lucas-Renato-Miguel-JogodaForca.c

Repare que o nome do arquivo fonte (.c) deve ter a mesma formatação do “Assunto” do e-mail: PE-T1-Lucas-Renato-Miguel-JogodaForca.c

- A primeira linha do código deve conter os **nomes dos componentes do grupo** e o **nome do exercício** que eles implementaram.
- Utilize nomes sugestivos para as variáveis. Faça corretamente a indentação e comentários no código para facilitar seu entendimento. Estes itens serão avaliados.
- A chave de fechamento de um comando deve estar na mesma direção do comando que fez a abertura. Por exemplo:

```
if(x > 0){  
  
} //O fechamento deve estar alinhado ao comando!!
```
- Preste atenção se está enviando a versão correta do trabalho, **visto que será considerada a data da entrega da versão correta.**
- Não deixe para enviar o trabalho na última hora, pois podem acontecer problemas com o envio.
- **Os trabalhos serão avaliados utilizando o Sistema Operacional Ubuntu 18.04.2 com gcc versão 7.4.0.**

- Trabalhos entregues com atraso receberão uma penalidade ***P***:

$$\mathbf{P} = 0,9 - 0,2(d - 1)$$

$$\mathbf{Nota\ Máxima} = \text{Valor do Trabalho} \times \mathbf{P}$$

onde d ($d \geq 1$) é o número de dias em atraso. Exemplo: se o grupo entregar com 1 dia de atraso, ele poderá ficar com, no máximo, 90% da nota do trabalho. Repare que após 5 dias de atraso não é mais possível entregar o trabalho. O atraso é contado em número de dias e não de horas. Caso o grupo envie o trabalho a qualquer hora do dia seguinte a data de entrega, por exemplo, será considerado 1 dia de atraso, e assim por diante.