



Aluno: _____ Nota: _____

1ª Especificação de Trabalho Prático

Instruções (leia com atenção):

As questões 1 a 7 devem ser feitas seguindo o modelo do programa corretor.c, aluno1.c, aluno2.h distribuído em conjunto com este trabalho. No arquivo aluno1.c tem as especificações de como cada questão deve ser preparada. No arquivo corretor.c já tem alguns exemplos de testes para as questões do trabalho.

As questões 8 e 9 devem ser feitas separadamente, com interação com o usuário.

Orientações de envio dos arquivos. Caso o trabalho esteja fora dessas orientações ele não será corrigido. Para entender as regras abaixo, suponha um aluno de nome João Silva, Matrícula: 2017902345

- 1) Crie uma pasta com o seguinte nome: NomeSobrenomeAluno–MatriculaAluno–T1. Para o exemplo de João seria: JoaoSilva–2017902345–T1. Todos os arquivos do trabalho devem estar dentro dessa pasta. Ao final, compacte ela e envie por e-mail o arquivo compactado anexado.
- 2) As seis questões q1 a q6 devem ser implementadas no arquivo aluno1.c. Esse arquivo deve ser renomeado para NomeSobrenomeAluno–MatriculaAluno–T1.c. Para o exemplo de Joao seria: JoaoSilva–2017902345–T1.c.

- 3) A questão q8 deve ser implementada em um arquivo com o nome NomeSobrenomeAluno–MatriculaAluno–Q8.c. Para o exemplo de Joao seria: JoaoSilva–2017902345–Q8.c.
- 4) A questão q9 deve ser implementada em um arquivo com o nome NomeSobrenomeAluno–MatriculaAluno–Q9.c. Para o exemplo de Joao seria: JoaoSilva–2017902345–Q9.c.

Questões

1) Faça um programa que valide uma data informada formato dd/mm/aaaa (d/m/aa, d/m/aaaa). As seguintes datas são válidas: 02/03/2015, 12/1/15, 1/9/2016. Considere ano bissexto.

2) Faça um programa em que o usuário informa a sua data de aniversário, o programa deve:

- a) Verificar se a data é válida (usar função da questão anterior);
- b) Quantidade de anos, meses, e dias, até a data atual

Entrada		Resultado			Observação
Data Inicial	Data Final	Anos	Meses	Dias	
06/06/2017	07/07/2017	0	1	1	
06/06/2017	05/07/2018	1	0	29	
26/07/2017	25/08/2017	0	0	30	São trinta dias de diferença, mas como julho tem 31 dias, não completou um mês.
26/06/2017	26/07/2017	0	1	0	Neste exemplo, são 30 dias de diferença também, mas neste caso completou um mês
27/02/2016	03/03/2017	1	0	4	

27/02/2015	03/03/2016	1	0	5	Neste caso, fevereiro de 2016 tem 29 dias
28/01/2016	29/02/2016	0	1	1	
28/02/2016	28/02/2017	1	0	0	São 366 dias de diferença, mas vamos assumir 1 ano
29/02/2016	28/02/2017	1	0	0	São 365 dias de diferença
29/02/2016	28/02/2020	3	11	30	
29/02/2016	29/02/2020	4	0	0	
29/02/2016	28/02/2019	3	0	0	

Não deve ser utilizada nenhuma função pronta na linguagem para calcular diferença entre datas. Exemplos de datas e resultado

3) Faça um programa que o usuário informa um texto de até 250, e uma letra. O programa deve informar quantas vezes essa letra aparece no texto. Desconsidere acentos nas vogais. Ou seja, um 'á' é igual a 'a'.

4) Faça um programa em que o usuário informa um texto de até 250, e uma palavra. O programa deve informar a posição inicial e fim de todas as ocorrências da palavra no texto.

5) Faça uma função que permita inverter um número inteiro N. Ex:

456 → 654

5430 → 345

100 → 1

6) Escreva uma função que determine quantas vezes um número K (de qualquer quantidade de dígitos) ocorre em um número natural N. Por exemplo:

O número 3 ocorre quatro vezes em 3539343.

O número 44 ocorre uma vez em 5444, e duas vezes em 54444

O número 23 ocorre duas vezes em 1234562354

7) Escreva uma função implemente o jogo caça palavras. Ela deve receber uma matriz de caracteres onde será feita a busca, e uma string a ser buscada. A função deve procurar a string em todas os sentidos possíveis: horizontal para frente e para trás, vertical para cima e para baixo, diagonal em todos os sentidos possíveis.

7) Fazer um programa que implementa o jogo da velha.

a) O programa deve sempre mostrar o tabuleiro do jogo da velha, com o jogador que deve jogar, da seguinte forma:

	1	2	3
A			
B			
C			

Jogador 2, informe sua jogada:

b) Cada jogador (jogador 1 e jogador 2) terá sua vez de jogar. O jogador um tem a marca “X” e o jogador 2 a marca “O”.

- c) O jogador deve informar a célula de interesse: ex: B3. O programa deve verificar se é uma célula válida
- d) O programa deve informar qual foi o ganhador, ou se não houve ganhador.

8) Faça um programa que implemente o jogo de batalha naval.

Ver o jogo em <http://www.ludijogos.com/multiplayer/batalha-naval/>

- a) O número de navios de guerra deve ser um parâmetro global. Precisando aumentar a quantidade de navios, o programador deve apenas mudar esse valor. b) Deve ter pelo menos 1 barco de tamanho 4, 1 de tamanho 3, 3 barcos de tamanho 1. Os demais barcos, se existirem devem ser obrigatoriamente de tamanho 2. Desta forma, o número mínimo de barco é: 5 barcos.
- c) O tabuleiro de cada jogador deve ser uma matriz 10 x 10. d) Cada jogador, no início do jogo deve dizer onde vai posicionar, cada um dos seus barcos. O programa deve controlar se os barcos estão posicionados de forma correta e) Use a seguinte configuração para representar o mapa:
 - i. [] espaço em branco -> posição do mar sem nada (e.g. barco, tiro do oponente, tiro do jogador).
 - ii. [N] Navio posicionado
 - iii. [O] Posição do navio abatida
 - iv. [X] Tiro inválido
- f) O programa deve ter uma função para o jogador poder Listar o tabuleiro.
 - i. Em seu tabuleiro ele deve ver seus barcos, os tiros do oponente e os espaços em branco.
 - ii. No tabuleiro do adversário deve ver seus tiros certos e seus tiros errados
- g) O programa deve ter a função para o jogador atirar, onde ele deve escolher uma célula da matriz para jogar.
- h) Cada jogador joga uma vez, depois é a vez do oponente.
- i) Vence o jogador que acertar todos os navios do oponente primeiro.