TRABALHO 2 - Caça palavras

ALOCAÇÃO DINÂMICA DE MATRIZES

Escreva um programa em C que implemente um jogo de caça-palavras. O programa deve representar uma matriz de caracteres (**alocada dinamicamente**) de dimensão *mxn* e buscar a ocorrência de palavras nesta matriz. As palavras podem estar na direção horizontal, vertical ou diagonal, em qualquer sentido (direto ou reverso). O programa deve:

1. ler a dimensão da matriz de caracteres: usuário digita a quantidade de linhas e colunas da matriz;

2. preencher a matriz: o preenchimento é feito da seguinte forma: o usuário digita um texto. Assuma que o usuário vai informar um texto que vai caber na matriz. O programa separa e armazena cada caractere do texto em uma posição da matriz, e quando uma linha da matriz é totalmente preenchida, o programa passa automaticamente a armazenar na próxima linha (e assim sucessivamente, até ter armazenado todo o texto digitado na matriz). Exemplo:

Entradas: tamanho da matriz: 5 x 9

texto digitado: "Esse é um texto usado apenas como exemplo da entrada"

matriz gerada e impressa: espaços em branco foram removidos do texto

Matriz:

	1	2	3	4	5	6	7	8	9
1	E	s	s	e	é	u	m	t	e
2	x	t	o	u	s	a	d	o	a
3	p	e	n	a	s	c	o	m	o
4	e	x	e	m	p	l	o	d	a
5	e	n	t	r	a	d	a		

3. imprimir a matriz;

4. buscar palavras na matriz: o programa lê palavras digitadas pelo usuário e realiza as buscas, imprimindo uma mensagem que diz se cada palavra ocorre ou não na matriz. Se ocorrer, o programa deve indicar:

- se a palavra ocorre na horizontal direta (da esquerda para direita), na horizontal reversa (da direita para esquerda), na vertical direta (de cima para baixo), na vertical reversa (de baixo para cima), na diagonal principal (direta ou reversa) ou na diagonal secundária (direta ou reversa);
- a posição na matriz do primeiro caractere da palavra encontrada e a posição na matriz do último caractere da palavra encontrada.

Exemplo:

	1	2	3	4	5	6	7	8	9
1	E	s	s	e	é	u	m	t	e
2	x	t	o	u	s	a	d	o	a
3	p	e	n	a	s	c	o	m	o
4	e	x	e	m	p	l	o	d	a
5	e	n	t	r	a	d	a	.	.

Palavra buscada: **usado**. Saída: palavra **usado** ocorre na horizontal direta, iniciando na posição [2,4] e terminando na posição [2,8]

Palavra buscada: **nxet**. Saída: palavra **nxet** ocorre na vertical reversa, iniciando na posição [5,2] e terminando na posição [2,2]

Palavra buscada: **oapd**. Saída: palavra **oapd** ocorre na diagonal principal direta, iniciando na posição [2,3] e terminando na posição [5,6]

OBS:
1) Havendo mais de uma ocorrência da palavra procurada, o programa pode mostrar apenas uma das ocorrências.
2) Lembre que a indexação de vetores e matrizes em C inicia na posição 0 (zero). Entretanto, a posição a ser mostrada dos caracteres na palavra encontrada deve iniciar em 1.

Observações para este trabalho:

- data da entrega: 12 de setembro;
- trabalhos atrasados não serão corrigidos - se você não terminou tudo a tempo, entregue o que conseguiu fazer;
- formato da entrega: enviar o arquivo fonte .c (não anexar arquivos executáveis!) para deise@inf.ufsm.br;
- trabalho em grupos de até 2 alunos. Enviar um e-mail por dupla, identificando a disciplina, nome dos alunos e número do trabalho (trabalho 2, por exemplo);
- caso não receba confirmação do recebimento do e-mail em 24h, contate novamente o professor (reclamações posteriores sobre possíveis problemas no envio do e-mail não serão aceitas);
- cópias da internet e/ou colegas de outras duplas anulam a nota do trabalho de todos os grupos envolvidos;
- trabalhos com erros de compilação não serão corrigidos;
- professor usará o Dev-C ou Code Blocks para corrigir os trabalhos. Podem usar qualquer IDE para implementar o trabalho, mas certifique-se de que o arquivo está compilando corretamente em alguma destas IDEs;
- obrigatoriamente a solução de vocês deve fazer uso de funções e passagem de parâmetros (eventuais respostas que pareçam corretas na execução, mas não usem funções e passagem de parâmetros, serão consideradas erradas);
- o *main* basicamente deve chamar funções e receber retorno de funções; toda a lógica da aplicação deve ser estruturada em outras funções;
  - espera-se, **ao menos**, a implementação das seguintes funções: alocação da matriz, preenchimento da matriz, impressão da matriz, busca de palavras e liberação da matriz. Evitem funções extremamente longas, como por exemplo, uma única função *busca* que faz todas as buscas na matriz em todas as direções. Sugestão: separar a busca em mais de uma função;
- não devem ser usadas variáveis globais;
- a matriz de caracteres **deve** ser alocada **dinamicamente** (baseada em vetor simples ou vetor de ponteiros);
- podem usar projetos, estruturando a aplicação em diversos arquivos. “Podem”, não é obrigatório! A partir do trabalho 3, o uso de projetos com múltiplos arquivos será obrigatório;
- a qualquer momento, vocês podem ser chamados para responder questionamentos sobre o trabalho entregue. Quando chamados, os alunos devem apresentar/responder aos questionamentos do professor, os quais farão parte da nota do trabalho;
- juntamente com o trabalho deve ser entregue um arquivo readme.txt que descreve brevemente o que foi implementado e o que não foi implementado do enunciado do trabalho;
- peso do trabalho: verificar no site da disciplina em “Avaliação”;