

Campus Apucarana**Disciplina: Fundamentos de Programação**

Exercícios para auxiliar nos estudos – Struct, funções, recursão, alocação dinâmica e arquivos

- *A lista pode ser feita em duplas ou trios.*
- *A solução dos exercícios deve ser entregue pelo Moodle.*
- *Enviar somente o código fonte (arquivo .c).*
- *Para as perguntas abertas, responder no próprio Moodle.*
- *Os códigos serão submetidas ao farejador de plágio. Soluções copiadas não serão pontuadas.*
- *Prazo para entrega: 11/05/21 (terça-feira) às 23h59*

1. Faça um programa que leia do teclado um cadastro de 10 alunos, indicando o nome, nota1, nota2. Calcule a média aritmética simples de cada um dos 10 alunos e, depois, escreva em um arquivo texto os dados de cada aluno: nome, nota1, nota2 e média. As notas e média deverão ser apresentadas como valores que possuem 2 casas decimais.
2. Faça um programa que leia os dados gerados pelo programa do exercício anterior e mostre na tela as informações referentes aos alunos. Calcule e mostre na tela a média geral da turma e o nome do aluno com a maior média.
3. Faça um programa que peça para o usuário fornecer uma palavra que deseja procurar e, em seguida, leia um arquivo texto do disco e procure por esta palavra no arquivo. Caso seja encontrada a palavra digitada pelo usuário em alguma das linhas do arquivo lido, o programa deverá exibir na tela o número da linha onde encontrou esta ocorrência, bem como o conteúdo da linha onde se encontra a palavra procurada.
4. Faça um programa que receba dois arquivos do usuário, e crie um terceiro arquivo com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo). Os nomes dos três arquivos devem ser fornecidos pelo usuário via console/terminal, ou seja, os nomes devem ser fornecidos como parâmetro para função main (vide conteúdo de argc e argv no slide de passagem de parâmetros).
5. Faça um programa que apresente um menu com as opções: adicionar um cliente, alterar informações do cliente, imprimir informações do cliente e remover um cliente. As informações de clientes que devem ser armazenadas são: nome, endereço, data de cadastro. Essas informações do cliente devem ser armazenadas em um arquivo binário. Lembre-se do conceito de persistência, portanto, a alteração das informações do cliente e a remoção também devem ser feitas no arquivo.

6. Escreva um programa que lê e modifica um arquivo texto trocando cada letra pela sua correspondente maiúscula. Exemplo: o arquivo abaixo deve ser modificado como a seguir:
Antes

```
a certificacao de metodologias que nos  
auxiliam a lidar com o julgamento imparcial  
das eventualidades aponta para a melhoria dos  
niveis de motivacao departamental.
```

Depois

```
A CERTIFICACAO DE METODOLOGIAS QUE NOS  
AUXILIAM A LIDAR COM O JULGAMENTO IMPARCIAL  
DAS EVENTUALIDADES APONTA PARA A MELHORIA DOS  
NIVEIS DE MOTIVACAO DEPARTAMENTAL.
```

7. Explique o funcionamento da função fseek. É mais fácil usá-la em arquivos no formato binário ou texto? Justifique a respostas.
8. Quando desejamos gravar os dados de uma struct em um arquivo, quais das funções é mais vantajosa: fprintf ou fwrite? E quando queremos gravar um vetor de struct? Justifique cada uma das suas respostas.
9. O que pode acontecer se encerrarmos um programa sem fechar um arquivo que foi aberto durante a execução desse programa?
10. Os números tetranacci iniciam com quatro termos pré-determinados e a partir daí todos os demais números são obtidos pela soma dos quatro números anteriores. Os primeiros números tetranacci são: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208... Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tetranacci.
11. Faça uma função recursiva que permita calcular a média de um vetor de tamanho N. O valor de N e de cada um dos elementos deve ser fornecido pelo usuário. Aloque dinamicamente um vetor para comportar esses valores.
12. Abra um arquivo texto, calcule e escreva o número de caracteres, o número de linhas e o número de palavras neste arquivo. Escreva também quantas vezes cada letra ocorre no arquivo (ignorando letras com acento). Obs.: palavras são separadas por um ou mais caracteres espaço, tabulação ou nova linha.

Desafio – Não vale nota!

Neste exercício você deverá implementar um quantificador de velocidade de digitação do usuário. Tome como exemplo o site: <https://www.typingtest.com/test.html>

Para este exercício é obrigatório o uso de alocação dinâmica. Use a menor quantidade de memória possível. Os textos usados para o teste de digitação também estão no Moodle da disciplina e são eles: 'digitacao1.txt' e 'digitacao2.txt'. O usuário deve escolher qual deles usar logo no início da execução do programa. O teste deve ter duração de 60 segundos. Não é permitido o uso de bibliotecas gráficas e o programa deve ser executado no console/terminal.

Ao final do teste, o algoritmo deve mostrar a velocidade de digitação do usuário (palavras por minuto), a quantidade de palavras digitadas erradas e a pontuação final, que é a quantidade de palavras corretamente escritas ao longo de um minuto. Em seguida, mostre ao usuário as palavras do texto original com cores. As palavras digitadas corretamente devem estar em verde enquanto as digitadas de maneira errada, em vermelho. Por conta da limitação do console/terminal, você pode optar por mostrar uma quantidade limitada de palavras, pedindo para o usuário digitar uma tecla específica para ver a segunda parte do texto. Pressionando novamente passa para a próxima parte até que se chegue ao final do DIGITADO PELO USUÁRIO. Observe que se o texto original possuir 15 parágrafos e usuário digitou apenas 2 desses parágrafos, somente esses 2 deverão ser mostrados no resultado.