

(arq_bin_01) Este exercício consiste em implementar funções para **salvar** e **ler** um vetor de um tipo específico **em um arquivo binário**.

Para isto, implemente uma estrutura simples que armazena os dados de um Aluno da UFES:

Strings: Nome, Data de Nascimento, Curso na UFES, Período de Ingresso,

Inteiro: % de conclusão

Float: Coeficiente de Rendimento Acadêmico.

Definição dos formatos de entrada e saída:

- **Entrada:**

Para testar sua implementação, faça um programa que leia o número de Alunos que serão armazenados, seguidos dos seus dados (na ordem apresentada acima), salvando-os em um vetor de Alunos dentro do programa.

Após finalizar a leitura dos alunos, **salve o vetor de alunos em um arquivo “alunos.bin”**.

Observações:

- Para os dados do tipo string, **salve o número mínimo de caracteres necessários** para armazenar o texto da variável. Ou seja, se você estiver usando `char texto[100]` para armazenar o texto “abc”, você deverá salvar apenas 3 elementos do tipo `char` (3 bytes). Utilize a função [fwrite](#) para isso.
- Após salvar todos os alunos, imprima a quantidade de bytes salvos no arquivo binário (seu navegador de arquivos deve mostrar o mesmo tamanho impresso pelo seu programa). Utilize o retorno da função `fwrite` para calcular a quantidade de bytes salvos no arquivo.

Após salvar os dados no arquivo, seu programa deve liberar toda a memória alocada para armazenar os dados dos alunos.

- **Saída:**

Em seguida, leia o vetor de alunos do arquivo “alunos.bin”, imprimindo os dados de todos os alunos.

Ver exemplos de formato de entrada e saída nos arquivos fornecidos com a questão.