

AULA PRÁTICA 12

- **Data de entrega: Até 13 de agosto às 23:59:59.**

- **Procedimento para a entrega:**

1. Submissão: via **Moodle**.
2. Os nomes dos arquivos e das funções devem ser especificados considerando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos *.h* e *.c* sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém *main()*, devem ser compactados (*.zip*), sendo o arquivo resultante submetido via **Moodle**.
6. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.
7. Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado.
8. Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas.
9. A avaliação considerará o tempo de execução e o percentual de respostas corretas.
10. Eventualmente, serão realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação.
11. Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada.
12. Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software.
13. Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota.
14. Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos.
15. Não serão considerados algoritmos parcialmente implementados.

- **Bom trabalho!**

Atividade

Tia Ivanilde é cunhada da Tia Andréa e também é uma respeitada professora, tem vários alunos e conhece o conceito de Tabelas *Hash*. Em sua última aula, ela prometeu que iria mostrar o resultado da criação de uma Tabela *Hash* usando **endereçamento aberto** com os dados do aluno. Contudo, ela não sabe implementar. Como você é um ótimo programador e já ajudou a Tia Andréa com árvores, você também se ofereceu para criar um programa para fazer isso para ela.

Sua tarefa é desenvolver o código e usar uma chave dupla na tabela *hash* considerando o nome e a matrícula (7 dígitos). A função $h(\cdot)$ usada deve seguir o padrão:

$$h(nome, idade) = \left(\sum_{i=0}^{|nome|-1} nome[i] * p[i \bmod |p|] + \sum_{i=0}^6 idade[i] \right) \bmod M$$

onde:

- p : é o vetor de pesos.
- $|p|$: é o tamanho do vetor de pesos.
- $|nome|$: é o tamanho da string com o nome.

- M : é o tamanho da tabela *hash*.
- mod: função de resto.

Considerações

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Uma tabela *hash* com endereçamento aberto deve ser criada, preenchida e percorrida para determinação da solução. **Nenhum algoritmo de ordenação deve ser utilizado.** Os procedimentos e o TAD *Hash* são similares aos vistos na aula teórica, porém, precisam ser adaptados para lidarem com uma chave dupla (nome e matrícula).

- Não altere o nome dos arquivos.
- O arquivo .zip deve conter na sua raiz somente os arquivos-fonte.
- Há vários casos de teste. Você terá acesso (entrada e saída) de casos específicos para realizar os seus testes.

Especificação da Entrada e da saída

A primeira linha sempre é a quantidade n de alunos. A segunda linha possui o tamanho p do vetor de pesos seguido pelos p pesos. Cada uma das próximas n linhas contém um nome (no máximo 50 caracteres), matrícula (inteiro), idade (inteiro) e peso (número real). **A sua Tabela *Hash* deve ser do tamanho n .**

Seu programa deve imprimir a Tabela *Hash* da primeira a última posição. Você deve imprimir o nome, matrícula, idade e peso seguindo o padrão abaixo.

| Entrada | Saída |
|---------------------------|---------------------------------|
| 3 | [2021235] daiane - 30 (50.0) |
| 5 1 3 5 2 1 | [2014381] pedro - 29 (90.0) |
| pedro 2014381 29 90.0 | [2213025] guilherme - 25 (70.0) |
| guilherme 2213025 25 70.0 | |
| daiane 2021235 30 50.0 | |

Diretivas de Compilação

```
$ gcc -c hash.c -Wall
$ gcc -c pratica.c -Wall
$ gcc hash.o pratica.o -o exe
```