

AULA PRÁTICA 2

- **Data de entrega: Até 21 de maio 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!**

Cursos da UFOP

Crie um programa que leia um caractere maiúsculo e um número inteiro. O caractere indica uma operação que deve ser realizada e o número inteiro (n) a quantidade de alunos de uma turma da UFOP. A turma T_n contém n alunos com *nome*, *curso* e *nota*. As operações são: *S* para soma e *M* para média. O aluno pode estar matriculado somente em um dos três cursos: *computacao*, *direito* e *nutricao*.

Calcule e mostre a soma ou a média por curso de todos os alunos. Ao final, libere a memória alocada para a turma e para os alunos.

Considerações

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Você deve criar dois TADS *Aluno* e *Turma* implementados em arquivos diferentes. Os TADs devem ser alocados e desalocados dinamicamente.

- 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 de entrada contém um único caractere Maiúsculo O ('S' ou 'M'), indicando a operação (Soma ou Média, respectivamente) que deverá ser realizada com os alunos da UFOP. A segunda linha corresponde ao tamanho da turma com n alunos e as próximas n linhas contém o nome (máximo de 20 caracteres), curso (máximo de 20 caracteres) e nota (valor real) de cada aluno.

A saída é a impressão do resultado solicitado (a soma ou média), com uma casa após o ponto decimal dos alunos de cada curso (*computacao*, *direito* e *nutricao*).

Entrada	Saída
S 200 Joao computacao 10.0 Carlos engenharia 9.0 Karla computacao 8.0 Fred nutricao 5.0 Flavia medicina 9.5 ...	A soma no curso de computacao eh 282.1 A soma no curso de direito eh 347.0 A soma no curso de nutricao eh 356.5

Diretivas de Compilação

```
$ gcc -c aluno.c -Wall  
$ gcc -c turma.c -Wall  
$ gcc -c pratica.c -Wall  
$ gcc aluno.o turma.o pratica.o -o exe
```

Avaliação de *leaks* de memória

Uma forma de avaliar se não há *leaks* de memória é usando a ferramenta valgrind. Um exemplo de uso é:

```
gcc -g -o exe *.c -Wall; valgrind --leak-check=yes -s ./exe < casoteste.in
```

Espera-se uma saída com o fim semelhante a:

```
==38409== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Para instalar no Linux, basta usar: `sudo apt install valgrind`.