

STL, Métodos de Ordenação, Bits

1. Escreva uma classe **Alunos** para gerenciar as notas de uma turma de alunos. Para isso, a entrada de dados deve ser da seguinte forma: o programa deve ler a matrícula de um aluno (inteiro), o nome de um aluno (string), duas notas (double) e em seguida passar para o próximo aluno. Esse processo de entrada de dados termina quando -1 for digitado para a matrícula de um aluno. Veja o exemplo abaixo:

Exemplo de execução do programa:

```
11
Tunico
3.5 5
12
Tinoco
10 8.5
-1
```

Em seguida, o programa desenvolvido por você deve imprimir na tela um relatório com as seguintes informações: a matrícula dos estudantes (em formato criptografado), o nome do estudante, e a média das notas dos alunos, ordenada da maior para a menor média, conforme exemplo de execução abaixo.

Exemplo de saída do programa:

```
0011 - Tinoco - 9.25
0100 - Tunico - 4.25
```

Para criptografia da matrícula do aluno você deve implementar um função que recebe um número inteiro e retorna o valor correspondente (em binário), invertido. Ou seja, o número 11, número de matrícula do aluno Tunico, é 1011 em binário. Logo, o retorno da sua função, para este caso, deve ser 0100.

Considerações Gerais!

- Exercício individual.
- Entrega: conforme agendado no PVANET Moodle;
- Crie um projeto para resolução do exercício. O projeto deve conter os arquivos `.h`, `.cpp`, e `main.cpp` criados para resolução do exercício. Envie, através do PVANet Moodle, uma pasta compactada (.rar ou .zip) contendo o projeto. A pasta compactada deve conter informações do aluno (ex.: julio_reis-pratica11.zip).
- O seu `main.cpp` deve conter, minimamente, instruções para criação (instanciação de objetos) e chamadas das funções implementadas (TODAS!!!). Para teste, você pode usar os exemplos fornecidos.