

# Trabalho Prático 01 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho  
Universidade Federal de Ouro Preto  
Departamento de Computação

7 de junho de 2021

## Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- 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 e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

## 1 roda.codigos

Então, tenho que admitir: o professor precisa da sua ajuda. Esse ano as coisas não estão correndo tão tranquilamente quanto eu queria, e não consegui finalizar o sistema do software das correções das atividades práticas a tempo. Uma parte vital está faltando, e como você sabe, nós precisamos desse sistema funcionando até esse mês, para avaliar os trabalhos práticos do período. A parte que está faltando é a que computa a pontuação dos alunos, dada a lista de submissões deles. Neste caso, a pontuação é relativa ao tempo necessário para acertar cada exercício: quem fizer mais rápido ganha mais pontos. Cada submissão errada de um exercício acarreta uma penalidade de 20 minutos, que só se concretizará se o exercício for acertado em algum momento.

Por favor, por favor, alguém me ajude!

### Especificação da Entrada

A primeira linha de caso de teste contém um único inteiro  $n$  indicando o número de submissões do caso de teste ( $1 \leq n \leq 300$ ). Cada uma das  $n$  linhas seguintes descrevem uma submissão; cada uma dessas linhas contém um identificador de problema (uma única letra entre 'A' e 'Z'), seguida por um inteiro  $t$  representando o instante de tempo em minutos ( $0 \leq t \leq 300$ ), seguido por um julgamento (a palavra “correto” ou a palavra “incompleto”). A entrada está em ordem crescente de tempo, e haverá no máximo um julgamento “correto” para cada problema.

### Especificação da Saída

Para cada caso de teste a entrada do seu programa deve imprimir uma linha contendo dois inteiros  $s$  e  $p$ , separados por um espaço, onde  $s$  é o número de problemas distintos com o julgamento “correto” e  $p$  é o somatório dos tempos nos quais cada problema distinto foi julgado pela primeira vez como “correto”, somado a 20 para cada julgamento “incompleto” recebido nesse problema, desde que no final o problema tenha sido julgado como “correto”.

### Exemplo de Entrada

```
5
A 100 correto
B 110 incompleto
B 111 correto
C 200 correto
D 300 incompleto
```

### Exemplo de Saída

```
3 431
```

### Estrutura do código

O código-fonte deve ser modularizado corretamente em três arquivos: principal.c, corretor.h e corretor.c. O arquivo principal.c deve apenas invocar as funções e procedimentos definidos no arquivo corretor.h. A separação das operações em funções e procedimentos está a cargo do aluno, porém, não deve haver acúmulo de operações dentro uma mesma função/procedimento.

Além disto, a informação de cada submissão deve ser armazenada em um tipo abstrato de dados criado especificamente para isso. Por fim, todas as informações das submissões de um caso de teste devem ser armazenadas em um vetor alocado dinamicamente (e posteriormente liberado) para cada caso de teste.

**Diretivas de Compilação**

```
$ gcc corretor.c -c  
$ gcc principal.c -c  
$ gcc corretor.o principal.o -o programa
```