

# Algoritmos e Estruturas de Dados 2

## Trabalho 1

Prof. Ricardo Oliveira - 2024/2

### Enunciado

Chandinho é um jovem antenado nas redes sociais. Depois que sua rede social favorita foi bloqueada por problemas judiciais, Chandinho resolveu criar sua própria rede social: o *Chuiter*.

O *Chuiter* é uma rede no estilo *micro-blog*. Nela, cada usuário pode fazer várias postagens, chamadas *chuite*. Cada chuite é um texto que pode ter no máximo 250 caracteres.

Chandinho tem um arquivo chamado **chuiter.txt** que armazena todos os chuites da rede, um por linha. Cada linha contém, nesta ordem: o *username* de quem fez a postagem; o *timestamp* indicando a data e hora da postagem, no formato UNIX<sup>1</sup>; e o texto da postagem em si.

Há um exemplo do arquivo **chuiter.txt** em anexo ao trabalho.

Escreva um programa em C que carrega os chuites do arquivo em uma estrutura de dados. Em seguida, seu programa deverá ler uma sequência de consultas do usuário. Cada consulta será digitada em uma única linha no formato

```
username dd/mm/yyyy hh:mm:ss DD/MM/YYYY HH:MM:SS
```

Para cada consulta, seu programa deverá listar, no formato exemplificado abaixo, todos os chuites feitos pelo usuário **username** entre os momentos **dd/mm/yyyy hh:mm:ss** e **DD/MM/YYYY HH:MM:SS**, inclusive.

Por exemplo, se o usuário digitar o comando

```
globo 08/07/2014 00:00:00 08/07/2014 23:59:59
```

seu programa deve listar todos os chuites feitos pelo usuário **globo** entre 00:00:00 de 08/07/2014 e 23:59:59 de 08/07/2014, inclusive.

Se o usuário digitar **\*** no lugar do **username**, seu programa deverá listar todos os chuites feitos por *todos* os usuários entre **dd/mm/yyyy hh:mm:ss** e **DD/MM/YYYY HH:MM:SS**, inclusive. Por exemplo, se o usuário digitar o comando

```
* 08/07/2014 00:00:00 08/07/2014 23:59:59
```

seu programa deve listar todos os chuites feitos (por todos os usuários) entre 00:00:00 de 08/07/2014 e 23:59:59 de 08/07/2014, inclusive.

Da mesma forma, se o usuário digitar **\* \*** no lugar de **dd/mm/yyyy hh:mm:ss**, os chuites deverão ser listados desconsiderando uma data/hora inicial; se o usuário digitar **\* \*** no lugar de **DD/MM/YYYY HH:MM:SS**, os chuites deverão ser listados desconsiderando uma data/hora final.

Por exemplo, se o usuário digitar o comando

```
meninonei * * 12/10/2010 14:42:00
```

seu programa deve listar todos os chuites feitos por **meninonei** até 14:42:00 de 12/10/2010. Ainda, se o usuário digitar o comando

---

<sup>1</sup>um inteiro indicando o número de segundos passados desde 1º de Janeiro de 1970

meninonei 01/01/2012 00:01:00 \* \*

seu programa deve listar todos os chuites feitos por **meninonei** a partir de 00:01:00 de 01/01/2012. Os usos de \* descritos acima podem ser combinados. Por exemplo, se o usuário digitar o comando

\* 02/07/2020 09:45:10 \* \*

seu programa deve listar todos os chuites feitos (por todos os usuários) a partir de 09:45:10 de 02/07/2020; se o comando for

monica \* \* \* \*

seu programa deve listar todos os chuites feitos por **monica**; e assim por diante.

Ao final de cada comando indique o número de chuites listados, conforme formato no exemplo abaixo.

A entrada termina quando o usuário enviar o sinal de EOF<sup>2</sup>.

Verifique o seguinte exemplo de execução (o sinal > indica entrada do usuário). Considere que foi carregado o arquivo **chuites.txt** de exemplo:

> globo 08/07/2014 00:00:00 08/07/2014 23:59:59

```
globo (08/07/2014 17:11:00): Alemanha abre o placar no Mineirao... Vamos, Brasil!!!
globo (08/07/2014 17:23:00): Mais um gol da Alemanha... #EuAcredito, Brasil!!! Vamos la!!!
globo (08/07/2014 17:24:00): Mais um gol da Alemanha... Nao desiste, Brasil! Vamos la!!!
globo (08/07/2014 17:26:00): Sou, brasileiro e nao desisto nunca!!! Vai, Brasil!!!
globo (08/07/2014 17:29:00): Quinto gol da Alemanha... Forca, Brasil!
globo (08/07/2014 18:24:00): Mais um gol pra Alemanha... #GloboNaCopa
globo (08/07/2014 18:34:00): Mais um pra Alemanha
7 chuites encontrado(s).
```

> \* 08/07/2014 00:00:00 08/07/2014 23:59:59

```
ocomentarista (08/07/2014 10:15:34): O Brasil tem tudo para fazer um grande jogo logo mais no Mineirao!
gbueno (08/07/2014 17:00:00): Autoriza o arbitro! Para cima deles Brasil!
globo (08/07/2014 17:11:00): Alemanha abre o placar no Mineirao... Vamos, Brasil!!!
globo (08/07/2014 17:23:00): Mais um gol da Alemanha... #EuAcredito, Brasil!!! Vamos la!!!
globo (08/07/2014 17:24:00): Mais um gol da Alemanha... Nao desiste, Brasil! Vamos la!!!
globo (08/07/2014 17:26:00): Sou, brasileiro e nao desisto nunca!!! Vai, Brasil!!!
gbueno (08/07/2014 17:26:10): Ta virando passeio
sbtofcial (08/07/2014 17:26:25): Pessoal, ta passando Chaves no SBT...
globo (08/07/2014 17:29:00): Quinto gol da Alemanha... Forca, Brasil!
globo (08/07/2014 18:24:00): Mais um gol pra Alemanha... #GloboNaCopa
globo (08/07/2014 18:34:00): Mais um pra Alemanha
gbueno (08/07/2014 18:45:12): Gool do Brasil. O chamado gol de honra da selecao brasileira.
ozagueiro (08/07/2014 19:05:00): Eu soh queria poder dar alegria para o meu povo :'(
13 chuites encontrado(s).
```

> meninonei \* \* 12/10/2010 14:42:00

```
meninonei (26/02/2002 10:36:07): To chegando com os refrii rapaziada !!
meninonei (26/11/2006 07:42:41): o ousado chegou !! kkkk
meninonei (02/05/2007 08:49:05): Acabo de completar "que famoso voce parece?" e meu resultado foi: kesh
3 chuites encontrado(s).
```

---

<sup>2</sup>Linux: Ctrl+D; Windows: Ctrl+Z, Enter

```
> meninonei 01/01/2012 00:01:00 * *
```

```
meninonei (05/09/2015 11:34:57): Hoje tem hein ... "cabelo de OUSADO, perfume de MALANDRO" ... Boa noite  
meninonei (23/06/2017 17:24:09): Ai galera tomem cuidado que tem um "orkut do bob sponja" se passando po  
meninonei (12/02/2017 04:59:41): Deus eh TOP  
3 chuites encontrado(s).
```

```
> * 02/07/2020 09:45:10 * *
```

```
alguemai (08/03/2023 09:27:15): Nem ia beber hoje mas slc, moh saudade do Alan Turing, programava muito  
rick (01/04/2024 00:00:01): https://www.youtube.com/watch?v=dQw4w9WgXcQ  
globo (04/05/2021 23:59:00): Mas a verdade tambem Ju, eh que voce nunca esteve sozinha  
monic (21/11/2023 05:44:22): #include <stdio.h> int main() { printf("Hello World!\n"); return 0; }  
4 chuites encontrado(s).
```

```
> monica * * * *
```

```
monica (05/08/2012 20:47:00): eu na antena parabolica  
monica (03/09/2009 22:55:12): ata  
2 chuites encontrado(s).
```

## Implementação

- Você pode escolher a(s) estrutura(s) de dados que achar melhor para sua implementação. Entretanto, você deve **obrigatoriamente** implementar *pelo menos* uma das seguintes estruturas:

- **Árvore AVL**
- **Árvore Rubro-Negra**
- **Árvore de Prefixo (trie)**

Outras estruturas podem ser utilizadas desde que *pelo menos* uma das dadas acima também seja;

- Você deve implementar pelo menos o método de **busca** da estrutura (para responder as consultas) e o método de **inserção de 1 (um) registro** da estrutura (para carregar o arquivo). Os chuites devem ser carregados, um por um, chamando esse método;
- A ordem em que os chuites são impressos em cada consulta não é relevante;
- O texto de cada chuite terá no máximo 250 caracteres. O *username* de cada usuário terá no máximo 25 caracteres;
- Cada *username* conterà apenas letras minúsculas;
- O texto de cada chuite pode conter qualquer caractere imprimível da tabela ASCII, incluindo espaços. O texto não terá caracteres fora da tabela (como acentos ou emojis);
- Lembre-se sempre de liberar toda a memória alocada ao final da execução do programa;
- Dica: para converter *timestamps* UNIX em data/hora legíveis e vice-versa, veja a `struct tm`, a função `mktime` e a função `localtime` da biblioteca `time.h`.

## Orientações

- O trabalho pode ser feito por equipes de *no máximo* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote **zip** ou **tar.gz** contendo todo o código-fonte necessário para compilar e executar sua solução, além de um arquivo de texto (txt) onde conste:

- O nome de todos os integrantes da equipe;
  - Qual(is) estrutura(s) de dados foi(ram) escolhida(s) pela equipe, e por quê;
  - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção;
  - **Atenção:** a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga **fielmente** o formato de entrada e de saída dado nos exemplos*, sob pena de grande redução da nota;
  - **Atenção:** Haverá *nota extra* de até 10 pontos (o trabalho pode valer 110 pontos ao todo) de acordo com o desempenho de seu programa. O trabalho **funcionando corretamente** mais eficiente (isto é, com menor tempo de execução para as entradas grandes usadas na correção) receberá 10 pontos de nota extra. Os demais trabalhos **funcionando corretamente** receberão nota extra proporcional em relação ao trabalho mais eficiente. As entradas grandes não serão fornecidas!
  - Certifique-se que seu programa compila e funciona antes de submetê-lo;
  - O trabalho deve ser entregue até **15 de Dezembro de 2024, 23:59**, apenas via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
  - Trabalhos detectados como cópia/plágio (de colegas, da internet ou de ferramentas de IA), ou comprados, receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.