

**Universidade de São Paulo**  
**Instituto de Ciências Matemáticas e Computação**  
**Departamento de Ciências de Computação**  
**SCC0201 - Introdução à Ciência de Computação II**

**2022**

**Professor:** Fernando Pereira dos Santos

**Monitores:**

Adrio Oliveira Alves

Gabriel Natal Coutinho

João Guilherme Jarochinski Marinho

**3º Trabalho Prático**

**Motivação:**

Criar um sistema de login com senha para que alunos possam checar suas notas na disciplina.

**Objetivo do trabalho:**

Desenvolva um programa em C capaz de:

- Cadastrar alunos através de seu Número USP (NUSP) e senha
  - Tentativas de cadastrar um NUSP mais de uma vez devem imprimir um erro e serem ignoradas pelo sistema.
- Realizar o login
  - Se efetuado com sucesso deve imprimir as notas atuais do aluno
  - Se houver erro, imprimir o tipo de erro (NUSP não cadastrado ou senha incorreta)

**Entrada:**

- Número de alunos a serem cadastrados (**int**)
- Alunos para o cadastro
  - NUSP (**string**), senha (**string**), notas (**4 doubles**). Uma por linha
- Número de tentativas de login (**int**)
- Logins
  - NUSP (**string**) e senha (**string**). Uma por linha.

**Saída:**

Durante o cadastro:

- “Cadastro efetuado com sucesso”
- “NUSP ja cadastrado”

Durante os logins:

- “Notas: P1=%1f, P2=%1f, T1=%1f, T2=%1f”
- “NUSP invalido”
- “Senha incorreta para o NUSP digitado”

**Exemplo:****Input:**

```
3
11444191 senha1 9.3 8.7 10.0 9.1
10621193 senha2 9.0 8.2 7.0 9.5
10621193 senha3 10.0 8.6 2.0 8.3
3
11111111 senha0
10621193 senha2
10621193 senha_errada
```

**Output:**

```
Cadastro efetuado com sucesso
Cadastro efetuado com sucesso
NUSP ja cadastrado
NUSP invalido
Notas: P1=9.0, P2=8.2, T1=7.0, T2=9.5
Senha incorreta para o NUSP digitado
```

## Detalhes:

1. Os alunos deverão ser cadastrados numa [Hash Table](#), onde a chave será o NUSP em questão.
  - a. O tratamento de colisões deverá ser resolvido através do encadeamento utilizando uma Linked List.
  - b. Outros detalhes da implementação da Hash Table ficam a cargo do aluno.
  - c. Dica: não crie uma HashTable com os 32 bits de endereçamento da hash, e sim manipule o hash para criar algo de um tamanho mais razoável (Ex: módulo 200 do hash, pegar 4 bits de cada ponta para formar um byte, etc...)
2. O armazenamento e checagem de senhas deverá ser protegido pelo mesmo algoritmo de hashing.
3. É obrigatória a utilização do seguinte algoritmo de hashing para uma entrada char \*S qualquer:
  - a. Criar dois unsigned ints de 16 bits (high = 0, low = 1)
  - b. Para cada caracter de S
    - i.  $low = (low + S[i]) \bmod 65521$
    - ii.  $high = (high + low) \bmod 65521$
  - c. Retornar um unsigned int de 32 bits, sendo que os 16 bits mais significativos correspondem a high e os 16 bits menos significativos correspondem a low.
4. A organização do código será levada em consideração para a avaliação.
5. Não plagie nem forneça seu código finalizado para colegas. Sujeito a nota 0 para todos os envolvidos.