Programação I (SIN110)

Prof.: Matheus Nohra Haddad

Trabalho Final - 20 pontos

DATA LIMITE DE ENTREGA: 13/05/2021

1 Contextualização

A testagem em massa é uma das medidas mais importantes e que possibilitam um maior controle sob a disseminação da pandemia de COVID-19.

Visando um melhor aproveitamento e gerenciamento dos testes de COVID-19, a UFV contratou seu grupo de programadores para criar um sistema que auxilie na gestão dos mesmos. A ideia é que seja possível ter um sistema capaz de cadastrar novos testes, armazenar informações sobre cada teste realizado e visualizar um relatório sobre todos os testes feitos.

2 Objetivo

Este projeto tem como objetivo o desenvolvimento de um sistema, em linguagem C, de gestão de testes para COVID-19. Seu programa será composto por um Menu de Opções nas quais cada uma delas terá um propósito. As opções estão exibidas e detalhadas a seguir:

```
-----------------------------------------------
         ##### TESTES COVID-19 #####

Menu de Opcoes:

1 - Cadastrar novos testes
2 - Consultar teste
3 - Cancelar teste
4 - Salvar informacoes em arquivo
5 - Visualizar informacoes de testes realizados
6 - Relatorio de testes realizados
0 - Sair
-----------------------------------------------
Digite a opcao:
```

Figura 1: Menu de Opções

0 - Cadastrar novos testes

Esta opção é responsável por cadastrar novos testes no sistema. Inicialmente será questionado quantos novos testes serão cadastrados, em seguida, o usuário deverá informar os seguintes dados, nesta ordem, para cada teste: nome completo, cpf, data de nascimento (dia, mês e ano), sexo (M ou F), bairro, resultado do teste (P - positivo ou N - negativo). A idade da pessoa também deverá ser armazenada, entretanto ela será calculada com base na data de nascimento e na data atual (obtida através da biblioteca time.h).

1 - Consultar teste

Esta opção é responsável por mostrar as informações de um teste cadastrado no sistema. Para isso, o programa deverá solicitar ao usuário que lhe informe o nome da pessoa a ser consultado e em seguida imprimir todas informações do teste desta pessoa na tela. Caso a pessoa ainda não tenha realizado nenhum teste, o programa deverá informar ao usuário.

2 - Cancelar teste

Esta opção é responsável por cancelar um teste realizado. Para isto, o usuário deve entrar com o número do teste a ser cancelado e em seguida, o programa deverá imprimir as informações relacionadas ao teste e perguntar se o usuário deseja realmente cancelar tal teste. Ao realizar esta tarefa, este teste deixará de ser válido. O cancelamento de testes deverá ser simulado através de uma string "valido"que indicará se o teste é válido ("SIM") ou foi cancelado ("NAO").

3 - Salvar informações em arquivo

Esta opção é responsável por salvar todas as informações dos testes em um arquivo. Para isto, o usuário deve informar o nome do arquivo .txt a ser salvo e o programa deverá criar o arquivo e salvar todas as informações neste arquivo. Somente testes válidos devem ser salvos.

4 - Visualizar informações de testes realizados

Esta opção é responsável por mostrar todas as informações armazenadas de todos os testes realizados, inclusive os que foram cancelados. Segue um exemplo na Figura 2:

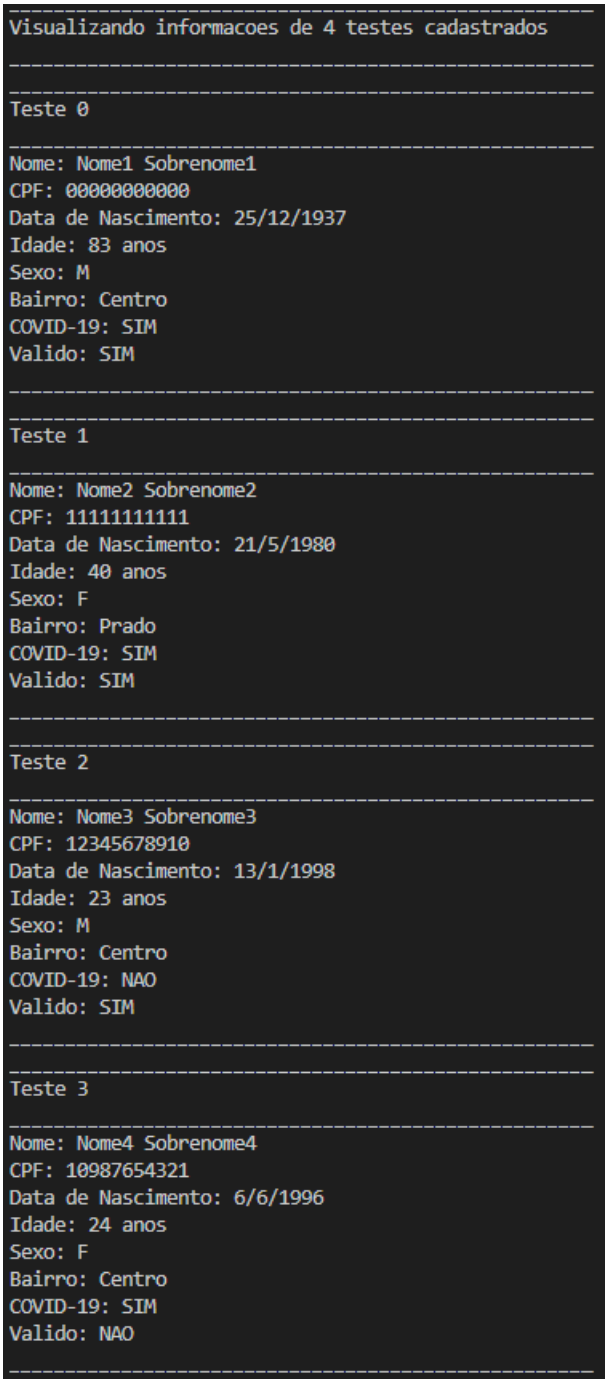

Figura 2: Informações sobre testes

5 - Relatorio de testes realizados

Esta opção é responsável por mostrar um relatório sobre os testes realizados. O programa deverá gerar um relatório com as seguintes informações: total de testes realizados, pessoas não infectadas (com porcentagem), pessoas infectadas (com porcentagem) – jovens, adultos e idosos infectados (com porcentagem). Pessoas são consideradas jovens se tiverem idade abaixo de 20 anos, adultos são considerados de 20 até 59 e idosos com mais de 60 anos. Segue um exemplo de relatório na Figura 3.

```
------------------------------------------------
RELATORIO COVID-19

------------------------------------------------
Total de pessoas cadastradas: 3
Pessoas nao infectadas: 1 (33.33%)
Pessoas infectadas: 2 (66.67%)
        Jovens infectados: 0 (0.00% dos infectados)
        Adultos infectados: 1 (50.00% dos infectados)
        Idosos infectados: 1 (50.00% dos infectados)
```

Figura 3: Relatório de testes

6 - Sair

Esta opção finalizará a execução do sistema.

3 Execução

O programa deverá se chamar "testesCovid"e poderá opcionalmente receber como parâmetro o nome de um arquivo .txt que já possui informações sobre testes realizados. Exemplo de chamadas possíveis para o programa:

- testesCovid.exe
- testesCovid.exe "testes.txt"

Obs: a funcionalidade de leitura de arquivo deverá **obrigatoriamente** ser implementada, apesar de ser uma forma alternativa de execução do programa.

3.1 Arquivo

As informações poderão ser lidas e/ou escritas em um arquivo .txt que deverá conter na primeira linha o número de testes cadastrados e nas linhas seguintes as informações de cada teste. Somente testes válidos serão lidos e/ou salvos no arquivo. Exemplo de arquivo:

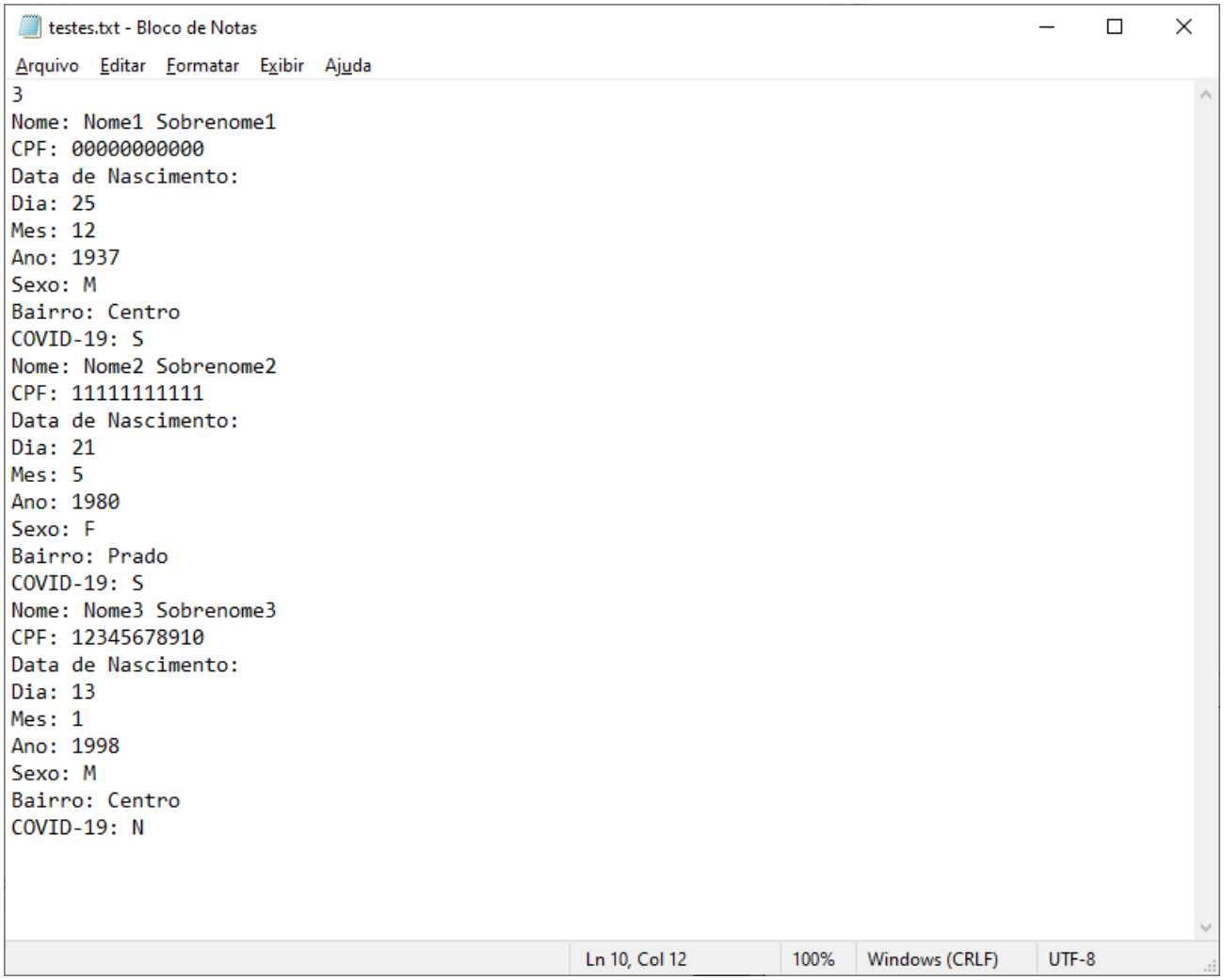

Figura 4: Arquivo com informações sobre testes

4 Avaliação

A Avaliação do Trabalho será feita através da análise do código fonte.

- **Análise do código fonte (CF):** feita obrigatoriamente por meio da documentação interna do código fonte, ou seja, por meio de comentários. Para cada linha de código do programa deverá ter uma linha de comentário, explicando o porque da utilização da instrução. O programa deverá possuir **obrigatoriamente**:

a) pelo menos uma estrutura (struct) criada pelos programadores;

b) pelo menos uma estrutura de seleção;

c) pelo menos uma estrutura de repetição;

d) pelo menos uma alocação dinâmica;

e) pelo menos duas funções.

OBS: OS ITENS ACIMA QUE NÃO FOREM RESPEITADOS SERÃO DESCONTADOS DA NOTA FINAL.

5 Considerações Finais

O projeto deverá ser feito em grupo de **até 3 integrantes** e submetido ao PVANET até o prazo final! Cada grupo deverá ter um único líder que será responsável pela submissão do projeto.

O líder do grupo deverá enviar um arquivo .zip (contendo todos os arquivos do trabalho final) com os números de matrícula dos integrantes do grupo (Exemplo: 0123_4567_8910.zip). Os nomes completos e as matrículas de todos os integrantes deverá constar no código-fonte na forma de comentário.
Além disso, é de extrema importância relevar as considerações a seguir:

- os trabalhos deverão ser entregues via **PVANET** até o prazo final. **Trabalhos enviados fora do prazo e/ou para o email do professor serão ignorados e receberão nota zero, impreterivelmente**.
- Caso o trabalho não siga as restrições descritas neste documento, ele será penalizado.
- **PLÁGIOS NÃO SERÃO TOLERADOS** e serão detectados por ferramentas de detecção automática e posteriormente serão comprovados pela análise visual do código fonte.
- O funcionamento correto do projeto é de inteira responsabilidade do grupo.
- O código fonte deverá ser enviado com a extensão *.c*. **Projetos com extensão .cpp serão avaliados com 50% de penalidade na nota final**.