**Instruções:**

1. O aluno deve acessar o Teams no dia e horário da prova para comprovar a sua presença;
2. O Aluno deve preencher o cabeçalho acima;
3. O Aluno deve postar este documento no Sava com o formato DOC;
4. O Aluno deve postar os arquivos da avaliação no GitHub (<https://github.com/>) e acrescentar link neste documento;
5. No dia da prova o professor poderá perguntar sobre alguns tópicos desta avaliação.

Casos de Exceção:

1. Caso o SAVA apresente problemas durante a avaliação, o PDF poderá ser enviado via Chat Privado do professor no Teams.
2. Caso o aluno tenha problemas com a conexão durante a avaliação, deverá entrar em contato com o professor imediatamente, para que ele tome as providências necessárias.

Link do trabalho no Github: <https://github.com/sylvioloureiro/Av1_Estrutura_De_Dados>

**Questão única:**

Você deve criar um sistema para gerenciar a fila de atendimento bancário, considerando os seguintes requisitos:

1- O programa deve permitir a entrada de dados dos seguintes campos:

1. Idade;
2. Cpf;
3. Sexo: 1- feminino e 2-masculino;
4. Para o sexo feminino responda a pergunta: Está grávida? 1-sim e 2-não;

2- O usuário deve escolher no menu a opção: Próximo atendimento e o programa deve exibir a próxima pessoa da fila de atendimento.

3- Casos excepcionais:

1. Todos os idosos (acima de 65 anos) têm prioridade na fila;
2. As mulheres grávidas têm prioridade na fila;
3. As prioridades entram na fila por ordem de chegada;

4- Após a chamada de atendimento os registros devem ser excluídos.

5- Crie um menu para listar as pessoas na fila.

Observação:

Somente pode ser usado listas duplamente encadeadas ou simplesmente encadeadas.

|  |  |
| --- | --- |
| Detalhamento da pontuação |  |
| Requisito 1 | 2 pontos |
| Requisito 2 | 2 pontos |
| Requisito 3 | 2 pontos |
| Requisito 4 | 2 pontos |
| Requisito 5 | 2 pontos |

#include <iostream>

using namespace std;

//variavis globais

int fnc=0;

int fpc=0;

int main() {

string cpf;

int idade;

string genero;

string fn[400];

string fp[400];

bool sair=1;

int menu;

while (sair==1){

cout << "\*\*\* EXISTEM " << fnc+fpc << " PESSOAS NA FILA \*\*\* " << "\n" << "\n" << "--------MENU--------" << "\n" << "1) Novo usuario" << "\n" << "2) Exibir fila" << "\n" << "3) Chamar proximo" << "\n" << "0) Sair" << "\n";

cin >> menu;

switch (menu) {

case 0:{ return 0;

}

case 1:{

int s=1;

while (s==1) {

int prioridade=2;

system ("CLS");

cout << "Informe CPF\n";

cin >> cpf;

cout << "Informe a idade\n";

cin >> idade;

cout << "Digite M para genero masculino e F para genero feminino\n";

cin >> genero;

if (genero=="f" or genero=="F") {

cout << "Gestante? Digite 1 para sim ou 2 para nao\n";

cin >> prioridade;

}

if (idade>64) {prioridade = 1;}

if (prioridade==2) {

fnc++;

fn[fnc-1] = cpf;

}

else {

fpc++;

fp[fpc-1] = cpf;

}

cout << "Deseja cadastrar um novo usuario? 1 para sim ou 2 para nao" << endl;

cin >> s;

system ("CLS");

}

break;

}

case 2:{

if(fpc>0){ for(int i=0;i<fpc;i++){

cout << fp[i] << " - Prioridade" << endl;}

}

if(fnc>0){ for(int i=0;i<fnc;i++){

cout << fn[i] << endl;}

}

break;

}

case 3:{

system ("CLS");

if(fpc>0){

cout << "----PROXIMO----: " << fp[0] << endl;

fpc--;

for(int i=0;i<fpc;i++){

fp[i]= fp[i+1];

}

}

else if (fnc>0){

cout << "CHAMANDO: " << fn[0] << endl;

fnc--;

for(int i=0;i<fnc;i++){

fn[i]= fn[i+1];

}

}

else cout << "FILA ZERADA" << endl << endl;

break;

}

}

}

return 0;

}