**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/Bardpedia/av1>

**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;

struct Lista {

int num;

Lista \*prox, \*ant;

};

int leiaIdade() {

int idade;

cout<<"\n Digite sua idade:"<<endl;

cin>>idade;

if (idade>65){

cout<<"\n Atentimento com Prioridade"<<endl;

}

return idade;

}

string leiaCpf(){

string cpf;

cout<<"\n Digite seu cpf:"<<endl;

cin>>cpf;

return cpf;

}

int leiaSexo(){

int sexo;

cout<<"\n Digite seu sexo 1 para Feminino 2 para Masculino: "<<endl;

cin>>sexo;

if (sexo==1){

cout<<"\n Esta gravida? 1-Sim ou 2-Nao"<<endl;

}

return sexo;

}

int main() {

Lista \*inicio = NULL, \*fim=NULL, \*aux = NULL;

int op=0, numero=0, achou = 0;

while(op != 5) {

cout<<"\n\t 1-Cadastrar Atendimento prioritario";

cout<<"\n\t 2-Cadastrar Atendimento";

cout<<"\n\t 3-Consultar a lista ";

cout<<"\n\t 4-Excluir registro de atendimento";

cout<<"\n\t 5-Sair";

cout<<"\n\t Escolha Opcao:"<<endl;

cin>>op;

if (op==1) {

Lista \*novo = new Lista;

novo->num = leiaIdade(),leiaCpf(),leiaSexo();

if (inicio == NULL) {

inicio = novo;

fim = novo;

cout<< "\nProximo Atendimento\n"<<novo;

novo->prox = NULL;

novo->ant = NULL;

}else {

novo->prox = inicio;

cout<<"+1 prox"<<inicio;

inicio->ant=novo;

cout<<"+1 ant"<<inicio;

novo->ant=NULL;

inicio=novo;

}

}

if (op==2) {

Lista \*novo = new Lista;

novo->num = leiaIdade(),leiaCpf(),leiaSexo();

if (inicio == NULL) {

inicio = novo;

fim = novo;

novo->prox = NULL;

novo->ant = NULL;

}else {

fim->prox = novo;

novo->ant=fim;

novo->prox=NULL;

fim=novo;

}

}

if(op ==3 ) {

if (inicio == NULL) {

cout<<"\nLista Vazia"<<endl;

}else {

aux = inicio;

while (aux != NULL){

cout<<"\n\tIdade:"<< aux->num <<endl;

cout<<"\n\tCPF:"<< aux->num <<endl;

cout<<"\n\tSexo:"<< aux->num <<endl;

aux=aux->prox;

}

}

}

if (op==4) {

if (inicio == NULL) {

cout<<"\nLista Vazia"<<endl;

}else {

aux = fim;//\*

while (aux != NULL){

cout<<"\n\tIdade:"<< aux->num <<endl;

cout<<"\n\tCPF:"<< aux->num <<endl;

cout<<"\n\tSexo:"<< aux->num <<endl;

aux=aux->ant;//\*

}

}

}

}

return 1;

}