**Solucionario del examen sustitutorio**

/\***Pregunta 1**.

Se tienen (ya existen) los archivos de texto NOTAS.TXT (que almacena: código del alumno (Cod), promedio de prácticas (PP),

examen parcial (EP) y examen final (EF) y SUSTITUTORIO.TXT (que almacena código del alumno (Cod) y examen sustitutorio(ES)),

de los cuales a continuación se presenta una parte de los mismos.

NOTAS.TXT

Cod PP EP EF

333 10.0 08 11

111 08.5 12 12

444 12.5 10 09

… … … …

SUSTITUTORIO.TXT

Cod ES

333 12

444 10

… …

Se pide crear un programa que empleando los archivos de texto NOTAS.TXT y SUSTITUTORIO.TXT, modifique (actualice)

el archivo NOTAS.TXT de forma que la nota del examen sustitutorio reemplace al examen parcial o al examen final

(según más convenga al alumno), sabiendo que el promedio final se calcula teniendo en cuenta que PP (peso 1),

EP (peso 1) y EF (peso 2).\*/

#include <iostream>

#include <fstream>

using namespace std;

//Nodo

struct Nodo{

//Campso de datos

int Cod;

float pp;

int ep;

int ef;

//Campo puntero

Nodo \*punt;

};

//Función que calcula promedio final

float promf(float pp,int ep,int ef){

return((pp+ep+ef\*2)/4.0);

}

//Función Crea\_Nodo

void Crea\_Nodo(Nodo \*&p,int Cod,float pp,int ep,int ef, int es,bool existe){

if(p==NULL){

p=new(Nodo);

p->Cod=Cod;

p->pp=pp;

p->ep=ep;

p->ef=ef;

if(existe){

if(promf(pp,es,ef)>promf(pp,ep,es)){

p->ep=es;

}

else{

p->ef=es;

}

}

p->punt=NULL;

}

else{

Crea\_Nodo(p->punt,Cod,pp,ep,ef,es,existe);

}

}

//Función que lee los archivos

void Lee(Nodo \*&p){

p=NULL;

int Cod1,Cod2,ep,ef,es;

float pp;

bool existe;

ifstream ent1,ent2;//Para abrir los archivos NOTAS.TXT y SUSTITUTORIO.TXT respec.

//Aperturando el archivo NOTAS.TXT

ent1.open("NOTAS.TXT");

if(!ent1){

cout<<"Error al abrir el archivo NOTAS.TXT para lectura...";

}

else{

while(!ent1.eof()){

if(ent1>>Cod1>>pp>>ep>>ef){

//Aperturando el archivo SUSTITUTORIO.TXT

ent2.open("SUSTITUTORIO.TXT");

if(!ent2){

cout<<"Error al abrir el archivo SUSTITUTORIO.TXT para lectura...";

}

else{

existe=false;

while(!ent2.eof() and !existe){

if(ent2>>Cod2>>es){

if(Cod1==Cod2){

existe=true;

}

}

}

//llamando a Crea\_Nodo

Crea\_Nodo(p,Cod1,pp,ep,ef,es,existe);

ent2.close();

}

}

}

ent1.close();

}

}

//Función que Modifica el archivo NOTAS.TXT

void Modifica(Nodo \*p){

ofstream sal;

sal.open("NOTAS.TXT");

if(!sal){

cout<<"Error al abrir el archivo NOTAS.TXT para modificarlo...";

}

else{

while(p!=NULL){

sal<<p->Cod<<"\t"<<p->pp<<"\t"<<p->ep<<"\t"<<p->ef<<endl;

p=p->punt;

}

sal.close();

cout<<"Archivo modificado..."<<endl;

}

}

//Función principal

main(){

Nodo \*Lista;

//llamando a Lee

Lee(Lista);

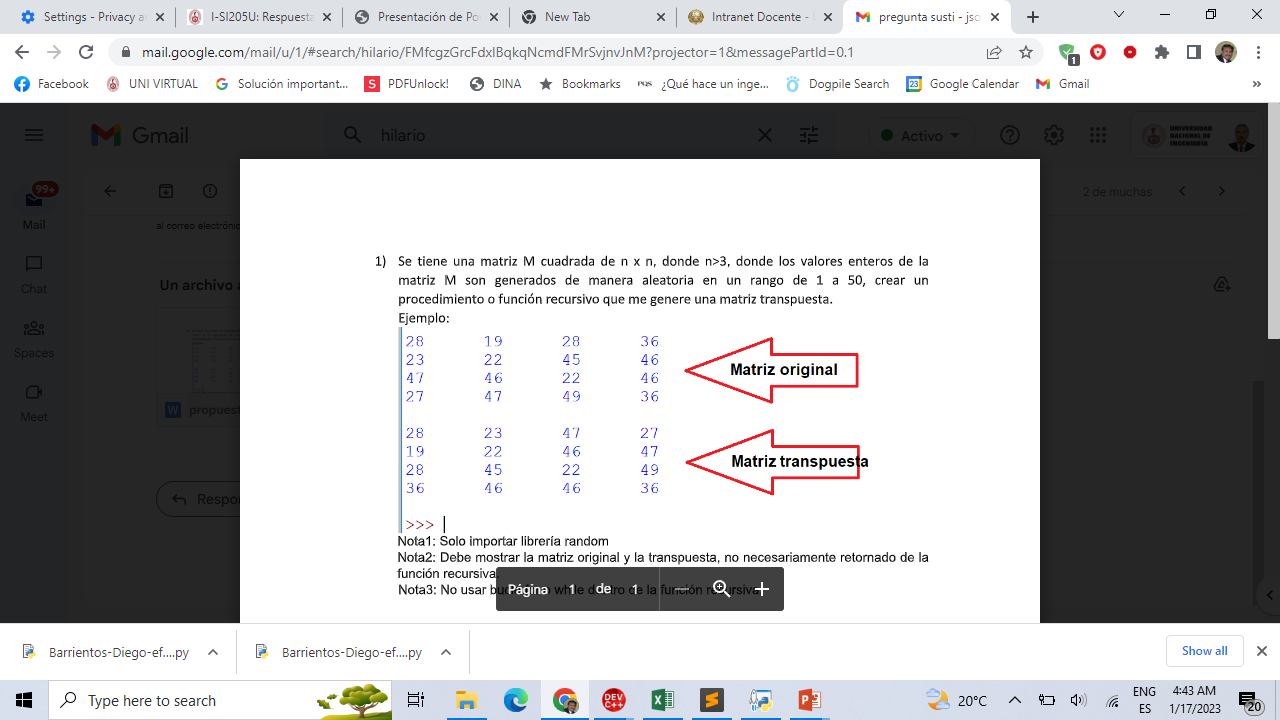
//llamando a Modifica

Modifica(Lista);

}

/\***Pregunta 2**.

Se tiene una matriz M cuadrada de n x n, donde n>3, y donde los valores enteros de la matriz M son generados de manera aleatoria en un rango de 1 a 50. Crear un procedimiento o función recursivo que genere una matriz transpuesta. Ejemplo:



Nota1: Solo importar librería random

Nota2: Debe mostrar la matriz original y la transpuesta, no necesariamente retornado de la función recursiva.

Nota3: No usar bucle for o while dentro de la función recursiva.\*/

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

void generarTranspuesta(int m[][50], int t[][50], int n, int i, int j) {

if (i == n) {

return;

}

if (j == n) {

generarTranspuesta(m, t, n, i + 1, 0);

return;

}

t[j][i] = m[i][j];

generarTranspuesta(m, t, n, i, j + 1);

}

int main() {

int n;

do{ cout << "Ingrese el tamaño de la matriz cuadrada (nxn): "; cin >> n;

if(n<4) cout<<"El valor de n debe ser mayor a 3, intente de nuevo"<<endl;

} while(n<4);

int matriz[50][50];

int transpuesta[50][50];

srand(time(NULL));

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

for (int j = 0; j < n; j++) {

matriz[i][j] = rand() % 50 + 1;

}

}

cout << "Matriz original:" << endl;

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

for (int j = 0; j < n; j++) {

cout << matriz[i][j] << " ";

}

cout << endl;

}

generarTranspuesta(matriz, transpuesta, n, 0, 0);

cout << "Matriz transpuesta:" << endl;

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

for (int j = 0; j < n; j++) {

cout << transpuesta[i][j] << " ";

}

cout << endl;

}

return 0;

}

/\***Pregunta 3**.

Escriba un programa que lea una oración y muestre las palabras ordenadas de mayor a menor número de caracteres. Ejemplo, si la oración ingresada es:

**La sonrisa cuesta menos que la electricidad pero da más luz**

Entonces el resultado sería:

**electricidad sonrisa cuesta menos pero que más luz La la da\*/**

#include<iostream>

#include<string.h>

using namespace std;

struct nodo{

string palabra;

nodo\* punt;

};

void creanodo(nodo\* &p, string pal){

if(p==NULL){

p=new(nodo);

p->palabra=pal;

p->punt=NULL;

}

else{

creanodo(p->punt,pal);

}

}

void recorre(nodo\* p){

if(p!=NULL){

cout<<p->palabra<<" ";

recorre(p->punt);

}

}

void ordena(nodo\* p){

while(p!=NULL){

nodo\* q=p->punt;

while(q!=NULL){

if(q->palabra.size()>p->palabra.size()){

string aux;

aux=p->palabra;

p->palabra=q->palabra;

q->palabra=aux;

}

q=q->punt;

}

p=p->punt;

}

}

int main(){

nodo\* p;

p=NULL;

string oracion,palabra;

cout<<"Oracion: ";getline(cin,oracion);

int i=0,tam=oracion.size();

while(i<tam){

while(oracion[i]==' ' and i<tam){

i++;

}

palabra="";

while(oracion[i]!=' 'and i<tam){

palabra=palabra+oracion[i];

i++;

}

creanodo(p,palabra);

}

ordena(p);

recorre(p);

return 0;

}