**UNIVERSIDAD NACIONAL DE INGENIERIA**

#### FACULTAD DE INGENIERIA INDUSTRIALA Y DE SISTEMAS

**Área Académica de Sistemas y Telemática**

#### CICLO: 2022-1

#### FECHA:26/07/2022

## CURSO: ALGORITMIA Y ESTRUCTURA DE DATOS (SI-205)

**RESOLUCIÓN DEL EXAMEN FINAL**

**NOTA:**

1. Responder cualquier pregunta de forma continua y completa. No se calificará trozos de respuestas en diferentes partes.
2. No hacer borrones en el limpio. Se descontará por borrones. Use su hoja de borrador o liquid.
3. Sin copias, apuntes. Solo use lapicero de tinta azul o negro.
4. Sírvase facilitar su identificación presentando su carné.
5. Se tiene (ya existe) el archivo de texto llamado ALGORITMIA.TXT, el cual contiene letras en mayúscula (que corresponde a las secciones) y valores numéricos (que corresponde a las notas de 0 a 20 del Examen Final). A continuación, se muestra una parte del archivo ALGORITMIA.TXT

W12 U15 W8 V11 U10

W11 U11 V6 V10 W11

... ... .. ... ...

De donde se puede observar que solo para dicha parte del archivo:

Las notas de la sección U son: 15, 10 y 11

Las notas de la sección V son: 11, 6 y 10

Las notas de la sección W son: 12, 8, 11 y 11

Diseñe un programa que empleando el archivo ALGORITMIA.TXT y listas enlazadas, muestre la sección o secciones con el mayor número de aprobados y el promedio de notas de dicha sección o secciones.

(7 puntos)

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

struct Nodo{

char sec;

int numaprobados,numalumnos;

float promedio;

Nodo \*punt;

};

void crear\_nodo(Nodo \*&p,char cod[]){

char sec;

sec=cod[0];

int num;

if(strlen(cod)==3){

num=(int)(cod[1]-48)\*10+(int)(cod[2]-48);

}

else{

num=(int)(cod[1]-48);

}

if(p==NULL){

p=new(Nodo);

p->sec=sec;

p->numalumnos=1;

if(num>=10){

p->numaprobados=1;

}

else{

p->numaprobados=0;

}

p->promedio=num;

p->punt=NULL;

}

else{

if(p->sec==sec){

p->promedio=(p->promedio\*p->numalumnos+num)/(p->numalumnos+1);

p->numalumnos+=1;

if(num>=10){

p->numaprobados+=1;

}

}

else{

crear\_nodo(p->punt,cod);

}

}

}

void leer\_archivo(Nodo \*&p){

p=NULL;

ifstream entrada;

entrada.open("ALGORITMIA.TXT");

if(!entrada){

cout<<"Error de apertura...";

}

else{

char cod[4];

while(!entrada.eof()){

if(entrada>>cod){

crear\_nodo(p,cod);

}

}

entrada.close();

}

}

void imprimir\_mayores(Nodo \*p){

Nodo \*act;

act=p;

int mayor=0;

while(act!=NULL){

if(p->numaprobados>mayor){

mayor=p->numaprobados;

}

act=act->punt;

}

cout<<"\n--Secciones con mayor numero de aprobados--\n";

cout<<"Numero de aprobados: "<<mayor<<endl;

cout<<"Seccion\tPromedio\n";

while(p!=NULL){

if(p->numaprobados==mayor){

cout<<" "<<p->sec<<"\t "<<p->promedio<<endl;

}

p=p->punt;

}

}

int main(){

Nodo \*lista;

leer\_archivo(lista);

imprimir\_mayores(lista);

return 0;

}

1. Escriba un algoritmo que lea de un archivo de texto los datos de venta por vendedor de una tienda de electrodomésticos con n vendedores (donde n > 10) y determine la bonificación de cada vendedor. La bonificación se calcula considerando un % sobre el monto vendido igual a 11 – x, donde x representa el lugar ocupado en el ranking de ventas. La bonificación solo se otorga a los 5 vendedores con mayores ventas. Si hay empate en el 5to lugar, son bonificados todos los que empatan. Cada línea del archivo contiene el monto total vendido y el nombre del vendedor (ver ejemplo). Su algoritmo debe mostrar el ranking de vendedores (de mayor a menor venta) indicando el nombre del vendedor, el monto vendido y el monto de bonificación asignada.

Ejemplo del archivo de texto:

42067.00 Julia Rios

15023.70 Javier Ramirez

18097.00 Luis Alvarez

........ ...............

........ ...............

........ ...............

48749.40 Juan Millones

35267.00 Mariela Fajardo

20527.50 Manuel Lopez

(7 puntos)

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

int len\_archivo(){

ifstream entrada;

entrada.open("VENTAS.TXT");

if(!entrada){

cout<<"Error...";

}

else{

float num;

char nom1[10],nom2[10];

int n=0;

while(!entrada.eof()){

if(entrada>>num>>nom1>>nom2){

n++;

}

}

entrada.close();

return n;

}

}

struct empleado{

char nom1[10],nom2[10];

float venta;

int pos;

};

void leer\_datos(){

int n=len\_archivo();

empleado arreglo[n],aux;

ifstream entrada;

entrada.open("VENTAS.TXT");

if(!entrada){

cout<<"Error...";

}

else{

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

entrada>>arreglo[i].venta>>arreglo[i].nom1>>arreglo[i].nom2;

}

entrada.close();

}

//ordenar

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(arreglo[i].venta<arreglo[j].venta){

aux=arreglo[i];arreglo[i]=arreglo[j];arreglo[j]=aux;

}

}

}

//Posicion

arreglo[0].pos=1;

for(int i=1;i<n;i++){

if(arreglo[i].venta==arreglo[i-1].venta){

arreglo[i].pos=arreglo[i-1].pos;

}

else{

arreglo[i].pos=i+1;

}

}

cout<<fixed<<setprecision(2);

//Imprimiendo

cout<<"Monto\t\tNombre\t\tPto\tBono\tPorce\n";

for(int i=0;arreglo[i].pos<=5;i++){

cout<<arreglo[i].venta<<" "<<arreglo[i].nom1<<"\t"<<arreglo[i].nom2<<"\t"<<arreglo[i].pos<<"\t"<<(arreglo[i].venta\*(11-arreglo[i].pos))/100<<"\t "<<(11-arreglo[i].pos)<<"%"<<endl;

}

}

int main(){

leer\_datos();

return 0;

}

1. Construya un programa que reciba una oración y determine si la oración es un palíndromo. Una oración es palíndroma si al invertirla, sin considerar los espacios en blanco y considerando todo en mayúsculas o minúsculas, da la misma oración. Debe utilizar una función recursiva para invertir la oración sin espacios en blanco. Ejemplo, si la oración ingresada es;

Yo hago Yoga hoy

Sin espacios en blanco y todo en minúsculas sería:

yohagoyogahoy

Al invertirla da

yohagoyogahoy

por tanto, es palíndroma

(6 puntos)

#include<iostream>

#include<string.h>

using namespace std;

void minusculas(char texto[]){

for(int i=0;texto[i]!='\0';i++){

if(65<=texto[i] && texto[i]<=90){

texto[i]+=32;

}

}

}

void eliminar\_espacios(char texto[]){

int i=0;

while(texto[i]!='\0'){

if(texto[i]==' '){

//eliminando

for(int k=i;texto[k]!='\0';k++){

texto[k]=texto[k+1];

}

}

else i++;

}

}

void invertir(char texto[],int n,int i=0){

if(i<n){

char aux;

aux=texto[i];

texto[i]=texto[n-1];

texto[n-1]=aux;

invertir(texto,n-1,i+1);

}

}

int main(){

char oracion[100],orac[100];

cout<<"--Analizando si una oracion es palindroma---\n";

cout<<"\nIngresar oracion: \n";gets(oracion);

minusculas(oracion);

eliminar\_espacios(oracion);

strcpy(orac,oracion);

cout<<"\nSin espacios en blanco y solo en minusculas seria: \n"<<oracion<<endl;

invertir(oracion,strlen(oracion));

cout<<"\nAl invertirla da:\n"<<oracion<<endl;

cout<<"\nPor ende, ";

if(strcmp(oracion,orac)==0){

cout<<"SI es palindroma";

}

else{

cout<<"NO es palindroma";

}

return 0;

}

**TIEMPO: 116 minutos**

**LOS PROFESORES**