**Solucionario del Examen Final**

**Pregunta 1:**

/\* Para un grupo de n (n<=30) alumnos, se desea almacenar para cada uno la siguiente información código (entero de 2 dígitos), 5 notas de práctica (enteros), se elimina la nota más baja, examen parcial (entero), examen final (entero), examen sustitutorio (entero), sabiendo que:

PF= (0.3PP+0.3EP+0.4EF) donde, PF = promedio final, ES=examen sustitutorio

PP= (p1+p2+p3+p4+p5-(menor nota))/4

La nota aprobatoria final es 11, además el Examen sustitutorio remplaza al examen Parcial o Final (el que convenga), se pide:

a) El promedio final, sin considerar el ES, indicando con un mensaje aprobado o desaprobado

b) El promedio Final considerando el ES, si el promedio final es menor que PF anterior, debe dejar el promedio final más alto. \*/

#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;

//

struct alum{

int cod;

int pc[5];

int ep;

int ef;

int es;

float pf;

};

//

int min(alum x[],int i){

int min=0;

for(int j=0;j<5;j++){

if(j==0){

min=x[i].pc[j];

}else{

(x[i].pc[j]<min)?(min=x[i].pc[j]):(min);

}

}

return min;

}

//

float prompc(alum x[],int i,int min){

float prom=0;

for(int j=0;j<5;j++){

prom+=x[i].pc[j];

}

prom=(prom-min)/4;

return prom;

}

//

//

int main(){

int n,cod,ep,ef,es,min1,pc;

float pf,pp,pF,pf1,pf2;

do{

cout<<"Nro de alumnos: ";cin>>n;

}while(!(0<n&&n<=30));

alum a[n];

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

cout<<"Alumno "<<i+1<<": "<<endl;

do{

cout<<"Cod: ";cin>>cod;

}while(!(10<=cod&&cod<=99));

//busqueda while();

cout<<"Practicas: "<<endl;

for(int j=0;j<5;j++){

do{

cout<<"PC"<<j+1<<": ";cin>>pc;

a[i].pc[j]=pc;

}while(!(0<pc&&pc<=20));

}

do{

cout<<"EP: ";cin>>ep;

}while(!(0<ep&&ep<=20));

do{

cout<<"EF: ";cin>>ef;

}while(!(0<ef&&ef<=20));

do{

cout<<"ES: ";cin>>es;

}while(!(0<es&&es<=20));

a[i].ep=ep;a[i].ef=ef;a[i].es=es;

}

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

cout<<"Alumno "<<i+1<<": "<<endl;

min1=min(a,i);

pp=prompc(a,i,min1);

//cout<<"PromPP"<<pp<<endl;

pf=a[i].ep\*0.3+pp\*0.3+0.4\*a[i].ef;

//Considera la sustitución que más le conviene al alumno

pf1=a[i].es\*0.3+pp\*0.3+0.4\*a[i].ef;

pf2=a[i].ep\*0.3+pp\*0.3+0.4\*a[i].es;

if(pf1>pf2){

pF=pf1;

}else{

pF=pf2;

}

cout<<"Promedio final: "<<pf<<endl;

if(pf>=11){

cout<<"Aprobado"<<endl;

}else{

cout<<"Desaprobado"<<endl;

}

if(pf<pF){

a[i].pf=pF;

}else{

a[i].pf=pf;

}

cout<<"Promedio Final: "<<a[i].pf<<endl;

}

return 0;

}

**Pregunta 2:**

/\* Se tiene (ya existe) el archivo de texto llamado NOTAS.TXT, el cual contiene las notas de prácticas de un conjunto de alumnos (por cada alumno: Código, nota de la PC1, nota de la PC2, nota de la PC3 y nota de la PC4). A continuación, se muestra una parte del archivo:

222 13 08 11 13

444 06 03 10 09

111 11 18 05 09

555 10 08 07 11

... .. .. .. ..

Diseñe un programa que permita modificar el archivo NOTAS.TXT, de forma que elimine las líneas correspondientes a los alumnos cuyo promedio de práctica es menor a 10. Nota: el promedio se calcula eliminado la nota más baja.\*/

#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

struct Nodo{

int cod;

int p1;

int p2;

int p3;

int p4;

float pp;

Nodo \*sig;

};

void CreaNodo(Nodo \*&p, int cod, int p1, int p2, int p3, int p4){

if(p==NULL){

p = new Nodo;

p->p1=p1;

p->p2=p2;

p->p3=p3;

p->p4=p4;

p->cod=cod;

p->sig=NULL;

}

else{

CreaNodo(p->sig, cod, p1, p2, p3, p4);

}

}

void Lee(Nodo \*&p){

p=NULL;

int cod, p1, p2, p3, p4;

ifstream ent;

ent.open("NOTAS.TXT");

if(!ent){

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

return;

}

while(ent >> cod >> p1 >> p2 >> p3 >> p4){

CreaNodo(p, cod, p1, p2, p3, p4);

}

ent.close();

}

void Calcula(Nodo \*&p){

if(p!=NULL){

int min=20;

if(p->p1<min) min=p->p1;

if(p->p2<min) min=p->p2;

if(p->p3<min) min=p->p3;

if(p->p4<min) min=p->p4;

p->pp=(p->p1+p->p2+p->p3+p->p4-min)/3.0;

return Calcula(p->sig);

}

}

void Sobreescribe(Nodo \*p){

ofstream sal;

sal.open("NOTAS.TXT");

for(Nodo \*px=p; px!=NULL; px=px->sig){

if(px->pp>10){

sal << px->cod << " " << px->p1 << " " << px->p2 << " " << px->p3 << " " << px->p4 << "\n";

}

}

sal.close();

}

int main(){

Nodo \*Lista;

Lee(Lista);

Calcula(Lista);

Sobreescribe(Lista);

cout << "Cambios realizados...";

return 0;

}

**Pregunta 3:**

/\* Escriba un algoritmo que lea n números enteros y los coloque en una lista, luego lea un numero x entero de 2 cifras y elimine de la lista aquellos elementos cuya suma de cifras es menor a x. Finalmente su algoritmo deberá calcular mediante una función el mínimo, el máximo y el promedio de los valores que quedaron en la lista.\*/

#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

struct Nodo{

int dato;

Nodo \*sig;

};

void CreaNodo(Nodo \*&p, int dato){

if(p==NULL){

p = new Nodo;

p->dato=dato;

p->sig=NULL;

}

else{

CreaNodo(p->sig, dato);

}

}

void Lee(Nodo \*&p){

p=NULL;

int n, d;

do{

cout << "Ingrese n: "; cin >> n;

}while(!(n>0));

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

cout << "Valor " << i << ": "; cin >> d;

CreaNodo(p, d);

}

}

void Elimina(Nodo \*&p, int x){

if(p==NULL) return;

int sum=0;

for(int ph=p->dato; ph!=0; ph/=10){

sum+=ph%10;

}

if(sum<x){

Nodo \*h=p;

p=p->sig;

delete(h);

return Elimina(p, x);

}

Elimina(p->sig, x);

}

void Recorre(Nodo \*p){

if(p!=NULL){

cout << p->dato << " ";

Recorre(p->sig);

}

else{

cout << endl;

}

}

void funcionpedida(Nodo \*p){

float prom = 0;

int cont = 0, min, max;

bool fst=true;

for(Nodo \*hx=p; hx!=NULL; hx=hx->sig){

if(fst){

min=hx->dato;

max=hx->dato;

prom+=hx->dato;

fst=false;

}

else{

prom+=hx->dato;

if(hx->dato<min) min = hx->dato;

if(hx->dato>max) max = hx->dato;

}

cont++;

}

cout << "Minimo: " << min << endl;

cout << "Maximo: " << max << endl;

cout << "Promedio: " << float(prom/cont) << endl;

}

int main(){

Nodo \*Lista;

Lee(Lista);

cout << "Lista generada: "<< endl;

Recorre(Lista);

int x;

cout << "Ingrese x: "; cin>>x;

Elimina(Lista, x);

cout << "Lista resultante:" << endl;

Recorre(Lista);

funcionpedida(Lista);

return 0;

}