**Solucionario del Examen Final de Algoritmia y Estructura de Datos**

**Pregunta 1:**

//Parte 1: Crear los archivos aprobados.txt y jalados.txt

#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

typedef char Cadena[11];

int main(){

int i,n;

ofstream sal\_A;//Para los aprobados

ofstream sal\_D;//Para los desaprobados

Cadena Cod;

int Nota;

sal\_A.open("aprobados.txt");

sal\_D.open("jalados.txt");

do{

cout<<"Ingrese el # de alumnos:"; cin>>n;

}while(!(n>0));

if(!sal\_A){

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

}

else{

if(!sal\_D){

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

}

else{

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

cout<<"Alumno #"<<i<<endl;

do{

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

}while(strlen(Cod)!=3);

do{

cout<<"Nota:"; cin>>Nota;

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

if(Nota>=10){

sal\_A<<Cod<<"\t"<<Nota<<endl;

}

else{

sal\_D<<Cod<<"\t"<<Nota<<endl;

}

}

sal\_D.close();

}

}

sal\_A.close();

cout<<"Proceso terminado..."<<endl;

return 0;

}

//Parte: presenta los códigos con la mayor nota y menor nota

#include <iostream>

#include <fstream>

using namespace std;

typedef char Cadena[11];

int main(){

int Nota;

Cadena Cod;

ifstream ent\_A;//para el archivo de aprobados

ifstream ent\_D;//para el archivo de desaprobados

int max=0;//para la myor nota de los aprobados

int min=20; //para la menor nota de los desaprobados

//Calculando la mayor nota

ent\_A.open("aprobados.txt");

if(!ent\_A){

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

}

else{

while(!ent\_A.eof()){

if(ent\_A>>Cod>>Nota){

if(Nota>max){

max=Nota;

}

}

}

ent\_A.close();

}

//Presentando los códigos de los alumnos con la máxima nota

ent\_A.open("aprobados.txt");

if(!ent\_A){

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

}

else{

cout<<"Los codigos de los alumnos con la mayor nota igual a "<<max<<endl;

while(!ent\_A.eof()){

if(ent\_A>>Cod>>Nota){

if(Nota==max){

cout<<Cod<<endl;

}

}

}

ent\_A.close();

}

//Calculando la menor nota

ent\_D.open("jalados.txt");

if(!ent\_D){

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

}

else{

while(!ent\_D.eof()){

if(ent\_D>>Cod>>Nota){

if(Nota<min){

min=Nota;

}

}

}

ent\_D.close();

}

//Presentando los códigos de los alumnos con la menor nota

ent\_D.open("jalados.txt");

if(!ent\_D){

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

}

else{

cout<<"Los codigos de los alumnos con la menor nota igual a "<<min<<endl;

while(!ent\_D.eof()){

if(ent\_D>>Cod>>Nota){

if(Nota==min){

cout<<Cod<<endl;

}

}

}

ent\_D.close();

}

return 0;

}

**Pregunta 2:**

#include <iostream>

#include <fstream>

using namespace std;

//NodoN

struct NodoN{

int Nota;

NodoN \*puntN;

};

struct NodoS{

char Sec;

int aprob;

float prom;

NodoS \*puntS;

NodoN \*puntN;

};

//Función Crea NodoN

void Crea\_NodoN(NodoN \*&p, int Nota){

if(p==NULL){

p=new(NodoN);

p->Nota=Nota;

p->puntN=NULL;

}

else{

Crea\_NodoN(p->puntN, Nota);

}

}

//Función Crea NodoS

void Crea\_NodoS(NodoS \*&p, char Sec, int Nota){

if(p==NULL){

p=new(NodoS);

p->Sec=Sec;

p->aprob=0;

p->prom=0;

p->puntS=NULL;

p->puntN=NULL;

Crea\_NodoN(p->puntN, Nota);

}

else{

if(p->Sec==Sec){

Crea\_NodoN(p->puntN, Nota);

}

else{

Crea\_NodoS(p->puntS, Sec, Nota);

}

}

}

//Función Aprob\_promN que calcula el # de aprobados por sección y su promedio

void Aprob\_promN(NodoN \*p, int &contA, float &prom){

int cont=0;

float s=0;

contA=0;

while(p!=NULL){

//calcula promedio

s=s+p->Nota;

cont++;

//cuenta el # de aprobados

if(p->Nota>=10){

contA++;

}

p=p->puntN;

}

prom=(s/cont);

}

//Función MayorS que nos el mayor número de aprobados

int MayorS(NodoS \*p){

int max=0;

while(p!=NULL){

Aprob\_promN(p->puntN, p->aprob, p->prom);

if(p->aprob>max){

max=p->aprob;

}

p=p->puntS;

}

return(max);

}

//Función RecorreS que presenta las secciones con el mayor # de aprobados y su promedio

void RecorreS(NodoS \*p, int max){

while(p!=NULL){

if(p->aprob==max){

cout<<p->Sec<<endl;

cout<<p->aprob<<"\t"<<p->prom<<endl;

}

p=p->puntS;

}

}

int main(){

ifstream ent;

NodoS \*Lista=NULL;

int Nota,max;

char Sec;

ent.open("NOTAS.TXT");

if(!ent){

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

}

else{

while(!ent.eof()){

if(ent>>Sec>>Nota){

Crea\_NodoS(Lista, Sec, Nota);

}

}

ent.close();

}

max=MayorS(Lista);

RecorreS(Lista, max);

return 0;

}

**Pregunta 3:**

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <string.h>

using namespace std;

typedef char cadena[80];

typedef char palabra[20];

int main(){

int t,n,m,i,r,num;

int npal[20];

for(int i=0;i<20;i++){

npal[i]=0;

}

cadena t1,t2;

cout<<"Ingrese el # de oraciones:"; cin>>num;

for(int k=0;k<num;k++){

fflush(stdin);

cout<<"Escriba la oracion #"<<k+1<<" acontinuacion:"<<endl; gets(t1);

cout<<endl;

strcpy(t2,t1);

n=strlen(t2);

m=0;

i=0;

while(i<n){

//avanza los espacios en blanco

while(i<n && t2[i]==' '){

i++;

}

if(i<n){

m++;

}

//avanza los caracteres de la palabra

while(i<n && t2[i]!=' '){

i++;

r++;

}

npal[r]++;

r=0;

}

}

for(int i=0;i<20;i++){

if(npal[i]>0){

cout<<npal[i]<<" palabra de "<<i<<" letra"<<endl;

}

}

return 0;

}