**PONTIFICIA UNIVERSIDAD CATOLICA MADRE Y MAESTRA**

****

**Nombre:**

Félix Alejandro Guzmán 2014-0565

**Materia:**

Programación 2

**Profesor:**

JOSÉ L. ALONSO O.

# Problema #1

#include <iostream>

#include <cstring>

#include <cstddef>

#include <stdio.h>

#include <vector>

#include <sstream>

#define size1 100

using namespace std;

vector<string> split(string str, char delimiter);

int revisar(string n, vector<string> nombrereyes);

int cantidad (int n, vector<string> suc, string nombre);

int indice(vector <string> nombrerey, string nombresuc);

int main()

{

int r =0, contval=0, s=0;

string reyes, sucesores;

cout << "Digite la cantidad de Reyes de una dinastia: ";

cin >> r;

cout << "Digite los nombres de los Reyes: ";

fflush(stdin);

getline(std:: cin,reyes);

cout << endl;

vector<string> rey = split(reyes, ' ');

for(int i = 0; i < rey.size(); ++i)

{

for(int j=i+1; j < rey.size(); j++)

{

if((rey[i].compare(rey[j])) != 0 )

{

contval++;

}

}

contval =0;

}

if(contval > 20)

{

cout << "Debe haber menos de 20 nombres distintos" << endl;

return 0;

}

cout << "Digite la cantidad de Sucesores: ";

fflush(stdin);

cin >> s;

int contr =0, conts=0;

string nombrere[20];

vector <string> nombrereyes;

vector <int> cantreyes;

cout << "Digite los nombres de los Sucesores: ";

fflush(stdin);

getline(std:: cin,sucesores);

vector<string> suce = split(sucesores, ' ');

for(int i = 0; i < rey.size(); ++i)

{

for(int j=0; j < rey.size(); j++)

{

if((rey[i].compare(rey[j])) == 0 )

{

contr++;

}

}

if(revisar(rey[i], nombrereyes) == 0)

{

nombrereyes.push\_back(rey[i]);

cantreyes.push\_back(contr);

}

contr =0;

}

cout << endl;

/\*

for(int i=0; i < nombrereyes.size(); i++)

{

cout << "Nombre: " << nombrereyes[i] << " cant: " << cantreyes[i] << endl;

}

\*/

cout << endl;

vector <string> nombresuce;

vector <int> cantsucerores;

for(int i=0; i < s; i++)

{

for(int j=0; j < s; j++)

{

if((suce[i].compare(suce[j])) == 0 )

{

conts++;

}

}

if(revisar(suce[i],nombresuce) == 0)

{

nombresuce.push\_back(suce[i]);

cantsucerores.push\_back(conts);

}

conts=0;

}

bool si = false;

for(int i = 0; i < suce.size(); ++i)

{

if(indice(nombrereyes, suce[i]) == 0 )

{

int n = indice(nombrereyes, suce[i]);

if(i == (suce.size()-1))

{

cout << cantreyes[n] + cantidad(i, nombresuce, suce[i])+1 << endl;

}

else

cout << cantreyes[n] + cantidad(i, nombresuce, suce[i]) << endl;

}

else

cout << cantidad(i,nombresuce,suce[i]) << endl;

}

}

int indice(vector <string> nombrerey, string nombresuc)

{

for(int i=0; i < nombrerey.size(); i++)

{

if(nombresuc.compare(nombrerey[i]) == 0)

{

return i;

}

}

return -1;

}

int cantidad (int n, vector<string> suc, string nombre)

{

int cont=0;

// cout << "Funcion Cantidad" << endl;

for(int i=0; i < suc.size(); i++)

{

if(suc[i].compare(nombre) == 0)

{

cont++;

}

// cout << "Nombre: " << nombre << " cont: " << cont << " valor de i: " << i << endl ;

}

return cont;

}

int revisar(string n, vector<string> nombres)

{

int val = 0;

for(int i=0; i < nombres.size();i++)

{

if(n.compare(nombres[i]) == 0)

{

val++;

}

}

if(val != 0)

{

return 1;

}

return 0;

}

vector<string> split(string str, char delimiter) {

vector<string> internal;

stringstream ss(str); // Turn the string into a stream.

string tok;

while(getline(ss, tok, delimiter)) {

internal.push\_back(tok);

}

return internal;

}

# Problema #2

#include <iostream>

using namespace std;

int prueba1( int val[10], int n, int hor, int minu, int seg);

int main()

{

int val[10] = {6,2,5,5,4,5,6,3,7,6};

int n=0, hor=0, minu=0, seg=0, part1h , part2h , part1min, part2min, part1seg, part2seg;

cout << "Digite la cantidad de segundos: ";

cin >> n;

hor = (int)(n / 3600);

minu = (int)((n - hor \* 3600) / 60);

seg = n - (hor \* 3600 + minu \* 60);

cout << "Horas: " << hor << endl;

cout << "Minutos: " << minu << endl;

cout << "Segundos: " << seg << endl;

part1h = (hor%100 - hor%10)/10;

part2h = hor%10;

part1min = (minu%100 - minu%10)/10;

part2min = minu%10;

part1seg = (seg%100 - seg%10) / 10;

part2seg = seg%10;

cout << "Parte 1 hora: " << part1h <<endl;

cout << "Parte 2 hora: " << part2h << endl;

cout << "Parte 1 minuto: " << part1min << endl;

cout << "Parte 2 minuto " << part2min << endl;

cout << "Parte 1 segundo: " << part1seg << endl;

cout << "Parte 2 segundo: " << part2seg << endl;

cout << endl;

cout << endl;

int p1 = prueba1( val, n, hor, minu, seg );

cout << endl;

cout << endl;

cout << "Los LED'S encendidos hasta " << n << " son: " << p1;

}

int prueba1( int val[10], int n, int hor, int minu, int seg)

{

int res =0, res1h =0, res2h =0, res1min =0, res2min =0, res1seg =0, res2seg=0, cont=0;

int part1h=0, part2h=0, part1min=0, part2min=0, part1seg=0, part2seg=0;

while(cont<=n)

{

hor = (int)(cont / 3600);

minu = (int)((cont - hor \* 3600) / 60);

seg = cont - (hor \* 3600 + minu \* 60);

part1h = (hor%100 - hor%10)/10;

part2h = hor%10;

part1min = (minu%100 - minu%10)/10;

part2min = minu%10;

part1seg = (seg%100 - seg%10) / 10;

part2seg = seg%10;

res1h += val[part1h];

res2h += val[part2h];

res1min += val[part1min];

res2min += val[part2min];

res1seg += val[part1seg];

res2seg += val[part2seg];

cout << "Valor de cont: " << cont << endl;

cont++;

}

res = res1h + res2h + res1min + res2min + res1seg + res2seg;

return res;

}

# Problema #3

#include <iostream>

using namespace std;

int espalindromico(int n);

int esprimo(int n);

int main()

{

int p=0, aux=0;

bool enc = false;

cout << "Digite un numero para buscar el siguiente primo que tambien sea palindromo: ";

do

{

cin >> p;

if(p < 1 && p > 1000000)

{

cout << "El rango del numero es entre 1 y 1,000,000. Intente de nuevo.";

}

}while( p < 1 && p >1000000);

/\*

if(esprimo(101))

{

cout << "Es primo " << endl;

}

if(espalindromico(101))

{

cout << "Es palindromo";

}

\*/

aux = p;

while(!enc && aux < 1000000)

{

//cout << "Valor de p: " << aux << endl;

if(esprimo(aux))

{

if(espalindromico(aux))

{

cout << "El numero es: " << aux ;

enc = true;

}

}

aux++;

}

cout << endl ;

}

int espalindromico(int n)

{

int inverso=0;

for (int i= n; i > 0; i/=10)

{

inverso = inverso \* 10 + i % 10;

}

if(n == inverso)

{

return 1;

}

return 0;

}

int esprimo(int n)

{

int cont =0;

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

{

if((n % i) == 0)

{

cont++;

}

}

if(cont > 2)

{

return 0;

}

return 1;

}

# Problema #5

#include <iostream>

using namespace std;

int main()

{

int totalvacas=0, pesocamion =0, valsum=0, cant1=0, menorvaca=0, sum=0, menorlech =0, menorindice=0;

cout << "Digite la cantidad de Vacas a la venta: ";

cin >> totalvacas;

cout << endl;

cout << "Digite el peso total que el camion puede llevar: ";

cin >> pesocamion;

int peso[totalvacas],leche[totalvacas], listselvac[cant1], listlech[cant1], indice[totalvacas];

for(int i=0; i < totalvacas; i++)

{

cout << "Digite el peso en kilogramos para la vaca #" <<i+1 << ": ";

cin >> peso[i];

cout << "Digite la produccion de leche en litros por dia para la vaca #" << i+1 <<": ";

cin >> leche[i];

indice[i] = i+1;

}

for(int j =0; j <totalvacas ;j++)

{

for(int ind =j+1;ind < totalvacas; ind ++)

{

if(peso[j] > peso[ind])

{

menorvaca = peso[j];

peso[j] = peso[ind];

peso[ind] = menorvaca;

menorlech = leche[j];

leche[j] = leche[ind];

leche[ind] = menorlech;

menorindice = indice[j];

indice[j] = indice [ind];

indice[ind] = menorindice;

}

}

}

float prom=0, prom1=0;

for(int j =0; j <totalvacas ;j++)

{

for(int ind =j+1;ind < totalvacas; ind ++)

{

prom = leche[j] / peso[j];

prom1 = leche[ind] / peso[ind];

if(prom < prom1)

{

menorvaca = peso[j];

peso[j] = peso[ind];

peso[ind] = menorvaca;

menorlech = leche[j];

leche[j] = leche[ind];

leche[ind] = menorlech;

menorindice = indice[j];

indice[j] = indice [ind];

indice[ind] = menorindice;

}

}

}

for(int ind = 0; ind < totalvacas; ind++)

{

valsum += peso[ind];

if(valsum <= pesocamion)

{

listselvac[cant1] = peso[ind];

listlech[cant1] = leche[ind];

cant1++;

}

}

cout << endl;

cout << " -- Vacas seleccionadas --" << endl;

for(int i =0; i < cant1; i++)

{

sum += listlech[i];

cout << "Vaca #" << indice[i] << " con peso: " << listselvac[i] << " y produccion de leche por dia: " << listlech[i] << endl;

}

cout << "La cantidad maxima que se puede obtener de leche es: " << sum;

cout << endl << endl;

cout << "Promedios " << endl;

for(int i=0; i < totalvacas; i++)

{

prom = leche[i] / peso[i];

cout << prom << endl;

}

}

# Problema #6

#include <iostream>

using namespace std;

int main()

{

int preciores = 80, preciocerdo = 60, grasares = 20, grasacerdo = 32;

float cantlibras=0, cantres=0, cantcerdo=0 ,preciototal=0, precioplibra =0;

cout << "Digite la canditdad de libras que desea ligar para el albondigon : ";

cin >> cantlibras;

cout << endl;

cantcerdo = (0.05\*cantlibras) / 0.12;

cantres = cantlibras - cantcerdo;

precioplibra = preciores \* (cantres/cantlibras) + preciocerdo \* (cantcerdo/cantlibras);

preciototal = preciores \* cantres + preciocerdo \* cantcerdo;

cout << "Cantidad de libras de Res es: " << cantres << endl;

cout << "Cantidad de libras de Cerdo es: " << cantcerdo << endl;

cout << "Cantidad de libra de Res en % " << (cantres/cantlibras)\*100 << endl;

cout << "Cantidad de libra de Cerdo en % " << (cantcerdo/cantlibras)\*100 << endl;

cout << "El precio total a pagar es: " << preciototal << endl;

cout << "El precio por libra a pagar es: " << precioplibra;

cout << endl;

}

# Problema #7

#include <iostream>

#include <stdio.h>

#define size 50

using namespace std;

int main()

{

char carac[26] = {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

char cadena[size];

int valor[26] = { 2, 22, 222, 3, 33, 333, 4, 44, 444, 5, 55, 555, 6, 66, 666, 7, 77, 777, 7777, 8, 88, 888, 9, 99, 999, 9999};

char ant, nue, siguient, ant1;

int val=0, valnuev=0;

cout << "Digite una cadena para saber su valor en numeros: ";

gets(cadena);

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

{

if(cadena[i] >= 'a' && cadena [i] <= 'z')

{

cadena[i] -=32;

}

if(cadena[i] != ' ')

{

for(int j=0; j < 26; j++)

{

ant = cadena[i];

nue = cadena[i-1];

siguient = cadena[i-1];

siguient = siguient -1;

ant1 = cadena[i];

ant1 = ant1 ;

if(ant == nue && cadena[i] == carac[j])

{

cout << ' ' <<valor[j] ;

}

else if(cadena[i] == carac[j])

{

cout << valor[j];

}

}

}

else

{

cout << ' ';

}

}

cout << endl;

//cout << ant;

// cout << ant1<< endl;

// cout << siguient;

// cout << nue;

}