



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA INDUSTRIAL

PRÁCTICA Nro. 2 DE ALGORITMOS Y ESTRUCTURAS DE DATOS 2014-2
20 Octubre 2014

TIEMPO: 1 Hr. 50Min.

Indicaciones

- 1) Resuelva cada pregunta en páginas diferentes.
- 2) Cualquier función adicional que requiera en su solución; debe escribirla necesariamente.
- 3) No se aceptarán reclamos en pruebas desarrollados con lápiz ni con correctores líquidos.
- 4) Queda prohibido el uso de celulares y/o cualquier equipo multimedio de comunicación.

PREGUNTA 1:

Dado una pareja de enteros positivos a y b , se llaman números amigos si la suma de los divisores propios de uno es igual al otro número y viceversa. Los divisores no deben incluir los números a y b . Por ejemplo sea $a = 220$ y $b = 284$, entonces se tiene que:

Los divisores propios de a son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, esta suma es 284

Los divisores propios de b son: 1, 2, 4, 71 y 142, esta suma es 220.

Entonces se concluye que a y b son números enteros amigos.

Se tiene la siguiente clase y la función principal. Se pide implementar sólo las funciones donde se indica puntaje.

```
#include<iostream>
using namespace std;
const int N = 50;
class amigos{
private:
    int n; /* Almacena la cantidad de números amigos encontrados al procesar la
funcion Generar */
    int A[N][2]; //Almacena las parejas de números amigos
public:
    amigos(); //Constructor por defecto
    ~amigos(); //Destructor
    friend void Generar(amigos &); /* Genera la matriz de números amigos, para ello
se ingresa 1 par de números, enteros, si son amigos, se almacena en la matriz, y
sino nuevamente solicita el ingreso de otro par de números.
Termina el proceso cuando ambos pares de números son iguales a 0. */ (3 pts)
    void Print(); /* Muestra la matriz de números amigos o en su defecto el mensaje
"no se generó la matriz." */ (1 pt)
    bool FindAmigos(int x, int y, int &k); /* Busca un par de numero amigos para
ver si está dentro de la matriz, si es así, retorna true y la posición, caso
contrario retorna false. */ (1 pt)
    void SetAmigos(int k); //Intercambia los números amigos de la posición k (1 pt)
    int GetAmigos(int k, int &z); //retorna los dos números amigos intercambiados (1 pt)
};

int main()
{
    amigos Q;
    int a,b,t,y;
    Generar(Q);
    Q.Print();    Lectura(a,b);
    if(Q.FindAmigos(a,b,t) == true)
    { Q.SetAmigos(t);
      Q.Print();
      cout<<"Primer numero amigo  :"<<Q.GetAmigos(t,y)<<endl;
      cout<<"Segundo numero amigo:"<<y<<endl;
    }
    else
        cout<<"El par de amigos no estan, se han separado"<<endl;
}
```


PREGUNTA 2:

Dado el siguiente código y una posible ejecución escriba las funciones donde se indica puntaje.

```
#include <iostream>
#include <string.h>
#include <stdio.h>
const int N = 255;

using namespace std;
bool EsVocal(char c); // 1 punto
int CuentaVocales(char cad[N]); // 1 punto
void PrintVectorCad(char s[N][N], int n);
void AnalizaPalabra(char s[], int &nc, int &nv); // 2 puntos
// tmc = tres o mas consonantes, dv = dos vocales
void VectorCadenas(char s[], char tmc[N][N], char dv[N][N]); // 3 puntos

int main()
{
    char cad[N], tmc[N][N], dv[N][N];
    cout<<"Ingrese una cadena:\n";    gets(cad);
    VectorCadenas(cad, tmc, dv);
    return 0;
}
```

PREGUNTA 3: (6 puntos)

Ordene manualmente el siguiente arreglo aplicando paso a paso el algoritmo de *Quicksort*:

23	45	67	40	90	31	84	60	44	15	77	57	66	47	134	121
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

■ FHE, ERL, HHL / 20 Octubre de 2014