Práctica Conjuntos

A) int unico(int A[],int B[],int n)

Crea una funcion que guarda en B los elementos de A sin repetición. La función retorna el número de elementos guardados en B. Use el siguiente código para evaluar su función:

#include <iostream>

#include <limits>

using namespace std;

void imprime(int v[],int n){

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

cout<<v[i]<<" ";

}

cout<<endl;

}

int unico(int A[],int B[],int n){

return -1;//rellena aqui

}

int main(){

int v[30],u[30];

for(int i=0;i<30;i++){

v[i]=rand()%21;

}

imprime(v,30);

int nu=unico(v,u,30);

imprime(u,nu);

}

B) int interseccion(int A[],int B[],int C[],int c)

Cree la función interseccion que guarda en C la intersección de los elementos de A y B.

Deberá usar la función *unico* creada anteriormente para usar el siguiente código con el que probar.

#include <iostream>

#include <limits>

using namespace std;

void imprime(int v[],int n){

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

cout<<v[i]<<" ";

}

cout<<endl;

}

int unico(int A[],int B[],int c){

return -1;

}

int interseccion(int A[],int na, int B[],int nb,int C[]){

return -1;

}

int main(){

srand(time(0));

int v1[10],v2[10],u1[10],u2[10],I[10];

for(int i=0;i<10;i++){

v1[i]=rand()%21;

v2[i]=rand()%21;

}

imprime(v1,10);

imprime(v2,10);

int nu1=unico(v1,u1,30);

imprime(u1,nu1);

int nu2=unico(v2,u2,30);

imprime(u2,nu2);

int ni=interseccion(u1,nu1,u2,nu2,I);

imprime(I,ni);

}

C) int difsimetrica(int A[],int B[],int C[],int c)

Cree la función difsimetrica que guarda en C la diferencia simetrica de los elementos de A y B. Deberá usar la función *unico* creada anteriormente para usar el siguiente código con el que probar.

#include <iostream>

#include <limits>

using namespace std;

void imprime(int v[],int n){

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

cout<<v[i]<<" ";

}

cout<<endl;

}

int unico(int A[],int B[],int c){

return -1;

}

int difsimetrica(int A[],int na, int B[],int nb,int C[]){

return -1;

}

int main(){

srand(time(0));

int v1[10],v2[10],u1[10],u2[10],D[10];

for(int i=0;i<10;i++){

v1[i]=rand()%21;

v2[i]=rand()%21;

}

imprime(v1,10);

imprime(v2,10);

int nu1=unico(v1,u1,30);

imprime(u1,nu1);

int nu2=unico(v2,u2,30);

imprime(u2,nu2);

int nd=difsimetrica(u1,nu1,u2,nu2,D);

imprime(D,nd);

}