

```

//Conjuntos

#include <stdio.h>
#include <conio.h>
#include <ctype.h>

//PROTOTIPOS
void visualizar (int *puntero, int LimInf, int LimSup);
int pertenece (int *puntero, int Elem, int LimInf, int LimSup);
void cargaElementos (int *puntero, int LimInf, int LimSup);
void interseccion (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup);
void resta (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup);
void unir (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup);

void main ()
{
    /* ..... PARTE DECLARATIVA ..... */
    int opc;
    /* opcion usuario en menu principal */
    char opc2;
    /* opcion usuario en sub-menu */

    int Conjuntos [3][100], Elemento;
    int Indice, Indice2, Indice3;
    int LimiteInferior= -100, LimiteSuperior= 100;
    int DatosErroneos = 0;

    /* ..... PARTE EJECUTIVA ..... */
    do{
autorrepetitivo */
        clrscr();
        /* limpia pantalla usuario */
        puts("\n\n\t\tCONJUNTOS\n");
        puts("\t\t1 - Definir el universo");
        puts("\t\t2 - Cargar elementos de los conjuntos...");
        puts("\t\t3 - Operaciones sobre los conjuntos...");
        puts("\t\t4 - Visualizar conjuntos...");
        puts("\t\t0 - SALIDA DEL PROGRAMA.");

        printf("\n\tTecllea tu opcion: ");
        scanf("%d", &opc);

        switch (opc)
        {
opciones menu PRINCIPAL */
            {

```

```

case 1: //DEFINIR EL UNIVERSO
{
clrscr();
do {
inferior y el limite superior: \n");
puts("\n\n\t\t\tDefine el universo de los conjuntos con el limite
puts("\n\t\t\tLimite inferior (número entero): ");
scanf ("%d", &LimiteInferior);
puts("\n\t\t\tLimite superior (número entero): ");
scanf ("%d", &LimiteSuperior);
if (LimiteInferior > LimiteSuperior)
{puts ("\n\t\t\tDatos incorrectos...");
DatosErroneos = 1;}
else
{puts ("\n\t\t\tDatos correctos...");
DatosErroneos = 0;}
}while (DatosErroneos == 1);
break;
} //Fin case 1

case 2: //CARGAR ELEMENTOS EN LOS CONJUNTOS
{
do {clrscr();
puts("\n\n\t\t\tDefinicion del universo \n"); //Submenu
puts("\t\t\tA - Elementos del conjunto A");
puts("\t\t\tB - Elementos del conjunto B");
puts("\t\t\tC - volver al menu principal");
printf("\n\t\t\tTeclea tu opcion: ");
opc2 = toupper(getche());

switch (opc2)
{
case 'A':
{
cargaElementos (&Conjuntos[0][0], LimiteInferior,
LimiteSuperior);
break;
} //Fin case A
case 'B':
{
cargaElementos (&Conjuntos[1][0], LimiteInferior,
LimiteSuperior);
break;
} //Fin case B
} //Fin switch opc2
} while (opc2!= 'C');

```

PASO3.C

```

Conjuntos[2][0]= LimiteSuperior+1;
break;
} //Fin case 2

case 3: //OPERACIONES SOBRE LOS CONJUNTOS
{
do {clrscr();
puts("\n\n\t\tOperaciones sobre conjuntos\n"); //Submenu
puts("\t\tA - Pertenece al conjunto A o B?");
puts("\t\tB - A uni#n B");
puts("\t\tC - A intersecci#n B");
puts("\t\tD - A resta B");
puts("\t\tE - Volver al menu principal");
printf("\n\tTecl#e tu opci#n: ");
opc2 = toupper(getche());

switch (opc2)
{
case 'A': //OPERACION PERTENECE
{
//pertenece a A?
do {
printf ("\n\tIntroduce el elemento a consultar (entre %d
y %d):\n", LimiteInferior, LimiteSuperior);
scanf ("%d", &Elemento);
}while (Elemento<LimiteInferior || Elemento>LimiteSuperior);
if (pertenece (&Conjuntos[0][0], Elemento, LimiteInferior,
LimiteSuperior))
{printf ("\n\t\tEl elemento %d PERTENECE al conjunto A",
Elemento);}
else
{printf ("\n\t\tEl elemento %d NO PERTENECE al conjunto
A", Elemento);}
//pertenece a B?
if (pertenece (&Conjuntos[1][0], Elemento, LimiteInferior,
LimiteSuperior))
{printf ("\n\t\tEl elemento %d PERTENECE al conjunto B",
Elemento);}
else
{printf ("\n\t\tEl elemento %d NO PERTENECE al conjunto
B", Elemento);}
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
break;
} //Fin case A

```

```

PASO3.C
case 'B': //OPERACION UNION
{
LimiteInferior, LimiteSuperior);
    unir (&Conjuntos[0][0], &Conjuntos[1][0], &Conjuntos[2][0],
//Visualizar soluci#n
puts ("\nCONJUNTO C:\n");
visualizar (&Conjuntos[2][0], LimiteInferior, LimiteSuperior);
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
break;
} //Fin case B

case 'C': //OPERACION INTERSECCION
{
&Conjuntos[2][0], LimiteInferior, LimiteSuperior);
//Visualizar soluci#n
puts ("\nCONJUNTO C:\n");
visualizar (&Conjuntos[2][0], LimiteInferior, LimiteSuperior);
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
break;
} //Fin case C

case 'D': //OPERACION RESTA
{
LimiteInferior, LimiteSuperior);
    resta (&Conjuntos[0][0], &Conjuntos[1][0], &Conjuntos[2][0],
//Visualizar soluci#n
puts ("\nCONJUNTO C:\n");
visualizar (&Conjuntos[2][0], LimiteInferior, LimiteSuperior);
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
break;
} //Fin case D
} //Fin switch opc2
} while (opc2!= 'E');
break;
} //Fin case 3

case 4: //VISUALIZAR LOS CONJUNTOS
{puts ("\nCONJUNTO A:\n");
visualizar (&Conjuntos[0][0], LimiteInferior, LimiteSuperior);
puts ("\nCONJUNTO B:\n");
visualizar (&Conjuntos[1][0], LimiteInferior, LimiteSuperior);
puts ("\nCONJUNTO C:\n");
visualizar (&Conjuntos[2][0], LimiteInferior, LimiteSuperior);

```

PAS03.C

```
        puts ("\n\t\tPulsa una tecla para continuar...");
        getch();
    } //Fin case 4

    } //Fin switch menu principal

    } while (opc != 0); //Fin do-while autorrepetitivo
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
}

void visualizar (int *puntero, int LimInf, int LimSup)
{int Ind;
for (Ind = 0;
    *(puntero+Ind) >=LimInf && *(puntero+Ind) <=LimSup && Ind < 100;
    Ind ++)
    {printf ("%4d", *(puntero+Ind));}
}

int pertenece (int *puntero, int Elem, int LimInf, int LimSup)
{ int *puntoFinal = puntero+100;
  for (;
    *puntero >= LimInf && *puntero <= LimSup && puntero < puntoFinal && *puntero != Elem;
    puntero++);
  if (*puntero == Elem)
    {return 1;} //El elemento PERTENECE al conjunto
  else
    {return 0;} //El elemento NO PERTENECE al conjunto
}

void cargaElementos (int *puntero, int LimInf, int LimSup)
{ int *puntoFinal = puntero+100;

  printf ("\n\tIntroduce elementos entre %d y %d. La carga de datos termina al introducir un elemento
erroneo:\n", LimInf, LimSup);
  do {
    scanf ("%d", puntero);
    puntero++;
  }while (*(puntero-1) >= LimInf && *(puntero-1) <=LimSup && puntero < puntoFinal);
}

void interseccion (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup)
{int *punteroAFin = punteroA+100;

  //Cada elemento de A lo comparamos con los de B
  for (;
    *punteroA>=LimInf && *punteroA<=LimSup && punteroA < punteroAFin;
```

PAS03.C

```

        punteroA++)
        {
            if (pertenece (punteroB, *punteroA, LimInf, LimSup))
                {
                    *punteroC = *punteroA;
                    punteroC++;
                }
        }
    *punteroC=LimSup+1;
}

void resta (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup)
{int *punteroAFin = punteroA+100;

    //Cada elemento de A lo comparamos con los de B
    for (;
        *punteroA>=LimInf && *punteroA<=LimSup && punteroA < punteroAFin;
        punteroA++)
        {
            if (!pertenece (punteroB, *punteroA, LimInf, LimSup))
                {
                    *punteroC = *punteroA;
                    punteroC++;
                }
        }
    *punteroC=LimSup+1;
}

void unir (int *punteroA, int *punteroB, int *punteroC, int LimInf, int LimSup)
{int *punteroTope = punteroA + 100;
    //los elementos de A los copiamos en C
    for (;
        *punteroA>=LimInf && *punteroA<=LimSup && punteroA < punteroTope;
        punteroA++, punteroC++)
        {
            *punteroC = *punteroA;
        }
    //los elementos de B los copiamos en C
    punteroTope = punteroB + 100;
    for (;
        *punteroB>=LimInf && *punteroB<=LimSup && punteroB < punteroTope;
        punteroB++, punteroC++)
        {
            *punteroC = *punteroB;
        }
    *punteroC = LimSup+1; //marca de fin de elementos
}

```