

```

//Conjuntos

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

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\tTecla tu opcion: ");
        scanf("%d", &opc);

        switch (opc)
        {
            case 1: //DEFINIR EL UNIVERSO
                {
                    clrscr();
                    do {
                        puts("\n\n\t\tDefine el universo de los conjuntos con el limite
inferior y el limite superior: \n");
                        puts("\n\t\tLimite inferior (número entero): ");
                        scanf ("%d", &LimiteInferior);
                    } while (LimiteInferior < -100 || LimiteInferior > 100);
                }
            default:
                continue;
        }
    } while (opc != 0);
}
/* bucle
/* menu PRINCIPAL */
/* opcion de usuario */
/*

```

```

                                PASO2.C
                                puts("\n\t\tLimite superior (número entero): ");
                                scanf ("%d", &LimiteSuperior);
                                if (LimiteInferior > LimiteSuperior)
                                    {puts ("\n\tDatos incorrectos...");
                                      DatosErroneos = 1;}
                                else
                                    {puts ("\n\tDatos correctos...");
                                      DatosErroneos = 0;}
                                }while (DatosErroneos == 1);
                                break;
                                }//Fin case 1

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

    switch (opc2)
    {
    case 'A':
        {
        Indice= 0;
        printf ("\n\tIntroduce elementos entre %d y %d. La carga de datos
termina al introducir un elemento erroneo:\n", LimiteInferior, LimiteSuperior);
        do {
            scanf ("%d", &Conjuntos[0][Indice]);
            Indice++;
        }while (Conjuntos[0][Indice-1] >= LimiteInferior &&
Conjuntos[0][Indice-1]<=LimiteSuperior && Indice < 100);
        break;
        }//Fin case A
    case 'B':
        {
        Indice = 0;
        printf ("\n\tIntroduce elementos entre %d y %d. La carga de datos
termina al introducir un elemento erroneo:\n", LimiteInferior, LimiteSuperior);
        do {
            scanf ("%d", &Conjuntos[1][Indice]);
            Indice++;
        }while (Conjuntos[1][Indice-1]>=LimiteInferior &&
Conjuntos[1][Indice-1]<=LimiteSuperior && Indice < 100);
        break;
        }
    }
}
}

```

```

                PASO2.C
            }//Fin case B

        }//Fin switch opc2
    } while (opc2!= 'C');
    Conjuntos[2][0]= LimiteSuperior+1;
    break;
} //Fin case 2

case 3:        //OPERACIONES SOBRE LOS CONJUNTOS
{
do {clrscr();
puts("\n\n\t\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\tTeclea 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);
for (Indice = 0;
Conjuntos[0][Indice]<=LimiteSuperior && Indice < 100 && Conjuntos[0][Indice] != Elemento;
Indice++);
if (Conjuntos[0][Indice] == Elemento)
{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?
for (Indice = 0;
Conjuntos[1][Indice]<=LimiteSuperior && Indice < 100 && Conjuntos[1][Indice] != Elemento;
Indice++);
if (Conjuntos[1][Indice] == Elemento)
{printf ("\n\t\tEl elemento %d PERTENECE al conjunto B",

```

PASO2.C

```

Elemento);}
B", Elemento);}

else
    {printf ("\n\t\tEl elemento %d NO PERTENECE al conjunto
    puts ("\n\t\tPulsa una tecla para continuar...");
    getch();
    break;
    }//Fin case A

case 'B': //OPERACION UNION
{
//los elementos de A los copiamos en C
for (Indice = 0, Indice3 = 0;
Conjuntos[0][Indice]<=LimiteSuperior && Indice < 100;
    Indice++, Indice3++)
    {Conjuntos[2][Indice3]= Conjuntos[0][Indice];}
//los elementos de B los copiamos en C
for (Indice2=0;
Conjuntos[1][Indice2]<=LimiteSuperior && Indice2 < 100;
    Indice2++, Indice3++)
    {Conjuntos[2][Indice3]= Conjuntos[1][Indice2];}
Conjuntos[2][Indice3]= LimiteSuperior +1; //marca de fin de
elementos
//Visualizar soluci#n
puts ("\nCONJUNTO C:\n");
for (Indice = 0;
Conjuntos[2][Indice]<=LimiteSuperior) && Indice < 100;
    Indice++)
    {printf ("%d ", Conjuntos[2][Indice]);}
puts ("\n\t\tPulsa una tecla para continuar...");
getch();
break;
}//Fin case B

case 'C': //OPERACION INTERSECCION
{
//Cada elemento de A lo comparamos con los de B
for (Indice = 0, Indice3 = 0;
Conjuntos[0][Indice]<=LimiteSuperior && Indice < 100;
    Indice++)
    {for (Indice2 = 0;
Conjuntos[1][Indice2]<=LimiteSuperior && Indice2 < 100 && Conjuntos[1][Indice2] != Conjuntos[0][Indice];

```

```

PASO2.C
    Indice2++);
    if (Conjuntos[1][Indice2] == Conjuntos[0][Indice])
        {Conjuntos[2][Indice3] = Conjuntos[0][Indice];
        Indice3++;
        }
    }
    Conjuntos[2][Indice3]=LimiteSuperior+1;
    //Visualizar soluci#n
    puts ("\nCONJUNTO C:\n");
    for (Indice = 0;
Conjuntos[2][Indice]<=LimiteSuperior) && Indice < 100;
        (Conjuntos[2][Indice]>=LimiteInferior &&
        Indice ++)
        {printf ("%d ", Conjuntos[2][Indice]);}
    puts ("\n\t\tPulsa una tecla para continuar...");
    getch();
    break;
} //Fin case C

case 'D': //OPERACION RESTA
{
//Cada elemento de A lo comparamos con los de B
for (Indice = 0, Indice3 = 0;
Conjuntos[0][Indice]<=LimiteSuperior && Indice < 100;
        Indice++)
        {for (Indice2 = 0;
Conjuntos[1][Indice2]<=LimiteSuperior && Indice2 < 100 && Conjuntos[1][Indice2] != Conjuntos[0][Indice];
        Indice2++);
        if (Conjuntos[1][Indice2] != Conjuntos[0][Indice])
            {Conjuntos[2][Indice3] = Conjuntos[0][Indice];
            Indice3++;
            }
        }
    Conjuntos[2][Indice3]=LimiteSuperior+1;
    //Visualizar soluci#n
    puts ("\nCONJUNTO C:\n");
    for (Indice = 0;
Conjuntos[2][Indice]<=LimiteSuperior) && Indice < 100;
        (Conjuntos[2][Indice]>=LimiteInferior &&
        Indice ++)
        {printf ("%d ", Conjuntos[2][Indice]);}
    puts ("\n\t\tPulsa una tecla para continuar...");
    getch();
    break;
} //Fin case D

```

```

                                PASO2.C
                                } //Fin switch opc2
                                } while (opc2!= 'E');
                                break;
                                }//Fin case 3

                                case 4: //VISUALIZAR LOS CONJUNTOS
                                {puts ("\nCONJUNTO A:\n");
                                for (Indice = 0; (Conjuntos[0][Indice]>=LimiteInferior &&
                                Conjuntos[0][Indice]<=LimiteSuperior) && Indice < 100; Indice ++)
                                {printf ("%d ", Conjuntos[0][Indice]);}
                                puts ("\nCONJUNTO B:\n");
                                for (Indice = 0; (Conjuntos[1][Indice]>=LimiteInferior &&
                                Conjuntos[1][Indice]<=LimiteSuperior) && Indice < 100; Indice ++)
                                {printf ("%d ", Conjuntos[1][Indice]);}
                                puts ("\nCONJUNTO C:\n");
                                for (Indice = 0; (Conjuntos[2][Indice]>=LimiteInferior &&
                                Conjuntos[2][Indice]<=LimiteSuperior) && Indice < 100; Indice ++)
                                {printf ("%d ", Conjuntos[2][Indice]);}
                                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();
                                }

```