

```
#include<stdio.h>
#include<stdlib.h>

void main(){
//-----

//VARIABLES A USAR:

int i,j,k,l,t;
int n,bandera=1;
int x,y;
int menor;
int *V,*Va; // Los usaremos como vectores de etiquetas.
int *registro; // registrará que vertices han sido pintados.
int **adya,**peso,**Arbol;

//-----

//RESERVA DE MEMORIA:

printf("Ingrese el numero de Vertices:\n");
scanf("%d",&n);

adya = (int **) malloc (n * sizeof(int *));
for (k=0 ; k < n ; k++) adya[k] = (int *) malloc (n * sizeof(int));

peso = (int **) malloc (n * sizeof(int *));
for (k=0 ; k < n ; k++) peso[k] = (int *) malloc (n * sizeof(int));

Arbol = (int **) malloc (n * sizeof(int *));
for (k=0 ; k < n ; k++) Arbol[k] = (int *) malloc (n * sizeof(int));

V = (int *) malloc (n * sizeof(int));

registro = (int *) malloc (n * sizeof(int));

Va = (int *) malloc (n * sizeof(int));
//-----

// INICIALIZACION DE CONJUNTOS Y MATRICES:

for (k=0 ; k < n ; k++) V[k]=k;
for (k=0 ; k < n ; k++) Va[k]=k;
for (k=0 ; k < n ; k++) registro[k]=0;
for (i=0 ; i < n ; i++) for (j=0 ; j < n ; j++) Arbol[i][j]=0;
//-----

// LECTURA DE LAS MATRICES:
printf("Ingrese elementos de la matriz adyacencia:\n");
for(i=0;i<n;i++) for(j=0;j<n;j++) scanf("%d",&adya[i][j]);
getchar();
printf("Ingrese elementos de la matriz de pesos:\n");
for(i=0;i<n;i++) for(j=0;j<n;j++) scanf("%d",&peso[i][j]);

//-----
```

```

//-----
// ALGORITMO:
printf("\n\nResultado del Algoritmo:\n");

while(bandera==1){
bandera=0;
for (k=0 ; k < n ; k++){ // Recorre todas las componentes con etiqueta k.
    menor=10000;
    for (i=0 ; i < n ; i++){
        if(V[i]==k){ // Evalua solo los vertices "i" que tienen la etiqueta k.

            // Calculamos la arista con menor peso,saliendo de cualquier vertice
            // de la componente k,salvando la posicion con las variables x e y.
            for(j=0;j<n;j++){
                if (adya[i][j]==1 && menor>peso[i][j] && V[i]!=V[j]){
                    menor=peso[i][j];
                    x=i;
                    y=j;
                }
            }
        }
    }
}
// Luego de tener una arista nueva,dos componentes se vuelven una,y *Va
// guarda la nueva configuracion de etiquetas:

//-> Si el vertice "y" ya ha sido pintado:
if(registro[y]==1){
// Pintamos todas las etiquetas que tienen valor V[x] con Va[y]
for(t=0;t<n;t++)if(Va[t]==V[x])Va[t]=Va[y];
}

//-> Si el vertice "y" aun no ha sido pintado:
else {
// Pintamos todas las etiquetas que tienen valor V[y] con Va[x]
for(t=0;t<n;t++)if(Va[t]==V[y])Va[t]=Va[x];
registro[y]=1; // Registramos que estamos pintando el vertice y.
}
Arbol[x][y]=Arbol[y][x]=1; // La arista se registra en nuestro arbol.
}

// Se actualiza el vector de etiquetas con el vector de etiquetas modificado:
for (k=0;k<n;k++)V[k]=Va[k];
// Los valores de registro vuelven a cero,listo para una nueva iteracion.
for (k=0;k<n;k++)registro[k]=0;

// Imprimimos todas las aristas de nuestro arbol:
printf("\nE={");
for(i=0;i<n;i++) for(j=i;j<n;j++) if(Arbol[i][j]==1) printf("{%d,%d}",i+1,j+1);
printf("}\n");

for(k=0;k<n-1;k++) if(V[k]!=V[k+1]) bandera=1 ;
// Si no se logra entrar a "if(V[k]!=V[k+1])" ni una sola vez,significara
// que todas las etiquetas tienen un solo valor,por lo cual bandera seguira
// con valor 0,terminando con el bucle while.
}

//-----
printf("\n");
} // FIN DE PROGRAMA.

```