

MATRIX multiplication

#include<stdio.h>

int main(){

int a[2][2], b[2][2], c[2][2], i, j;

int m1, m2, m3, m4 , m5, m6, m7;

printf("Enter the 4 elements of first matrix: ");

for(i = 0;i < 2; i++)

for(j = 0;j < 2; j++)

scanf("%d", &a[i][j]);

printf("Enter the 4 elements of second matrix: ");

for(i = 0; i < 2; i++)

for(j = 0;j < 2; j++)

scanf("%d", &b[i][j]);

printf("\nThe first matrix is\n");

for(i = 0; i < 2; i++){

printf("\n");

for(j = 0; j < 2; j++)

printf("%d\t", a[i][j]);

}

printf("\nThe second matrix is\n");

for(i = 0;i < 2; i++){

printf("\n");

for(j = 0;j < 2; j++)

printf("%d\t", b[i][j]);

}

m1= (a[0][0] + a[1][1]) \* (b[0][0] + b[1][1]);

m2= (a[1][0] + a[1][1]) \* b[0][0];

m3= a[0][0] \* (b[0][1] - b[1][1]);

m4= a[1][1] \* (b[1][0] - b[0][0]);

m5= (a[0][0] + a[0][1]) \* b[1][1];

m6= (a[1][0] - a[0][0]) \* (b[0][0]+b[0][1]);

m7= (a[0][1] - a[1][1]) \* (b[1][0]+b[1][1]);

c[0][0] = m1 + m4- m5 + m7;

c[0][1] = m3 + m5;

c[1][0] = m2 + m4;

c[1][1] = m1 - m2 + m3 + m6;

printf("\nAfter multiplication using Strassen's algorithm \n");

for(i = 0; i < 2 ; i++){

printf("\n");

for(j = 0;j < 2; j++)

printf("%d\t", c[i][j]);

}

return 0;

}

2. merge sort

#include <stdio.h>

int main()

{

int n1,n2,n3;

int a[10000], b[10000], c[20000];

printf("Enter the size of first array: ");

scanf("%d",&n1);

printf("Enter the array elements: ");

for(int i = 0; i < n1; i++)

scanf("%d", &a[i]);

printf("Enter the size of second array: ");

scanf("%d",&n2);

printf("Enter the array elements: ");

for(int i = 0; i < n2; i++)

scanf("%d", &b[i]);

n3 = n1 + n2;

for(int i = 0; i < n1; i++)

c[i] = a[i];

for(int i = 0; i < n2; i++)

c[i + n1] = b[i];

printf("The merged array: ");

for(int i = 0; i < n3; i++)

printf("%d ", c[i]);

printf("\nFinal array after sorting: ");

for(int i = 0; i < n3; i++){

int temp;

for(int j = i + 1; j < n3; j++) {

if(c[i] > c[j]) {

temp = c[i];

c[i] = c[j];

c[j] = temp;

}

}

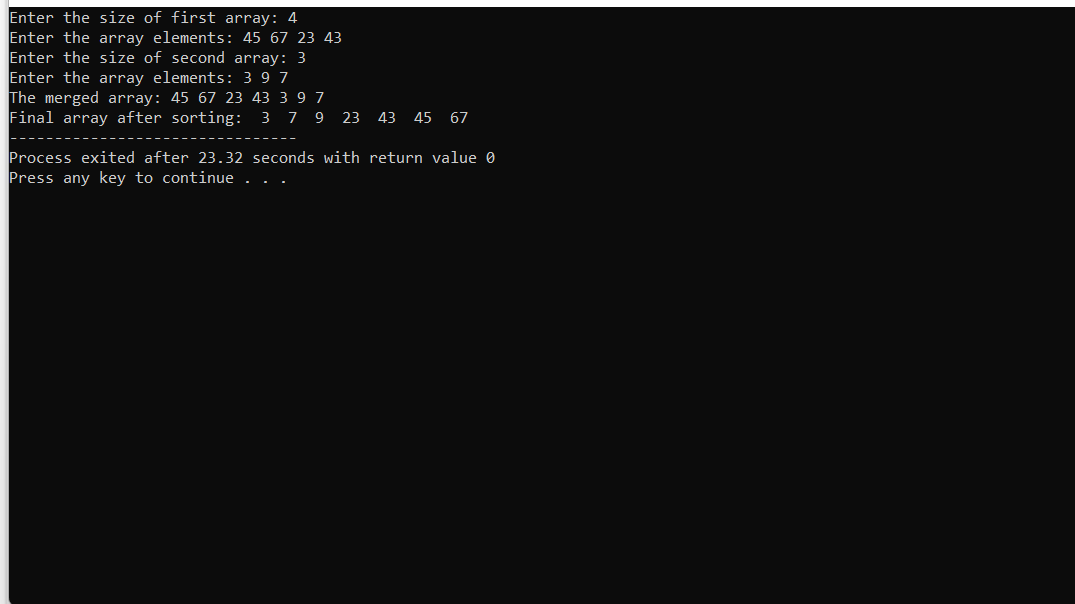
}

for(int i = 0; i < n3 ; i++)

printf(" %d ",c[i]);

return 0;

}



3. napsak problem

#include<stdio.h>

int max(int a, int b)

{

return (a > b)? a : b;

}

int knapSack(int W, int wt[], int val[], int n)

{

int i, w;

int K[n+1][W+1];

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

{

for (w = 0; w <= W; w++)

{

if (i==0 || w==0)

K[i][w] = 0;

else if (wt[i-1] <= w)

K[i][w] = max(val[i-1] + K[i-1][w-wt[i-1]], K[i-1][w]);

else

K[i][w] = K[i-1][w];

}

}

return K[n][W];

}

int main()

{

int i, n, val[20], wt[20], W;

printf("Enter number of items:");

scanf("%d", &n);

printf("Enter value and weight of items:\n");

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

scanf("%d%d", &val[i], &wt[i]);

}

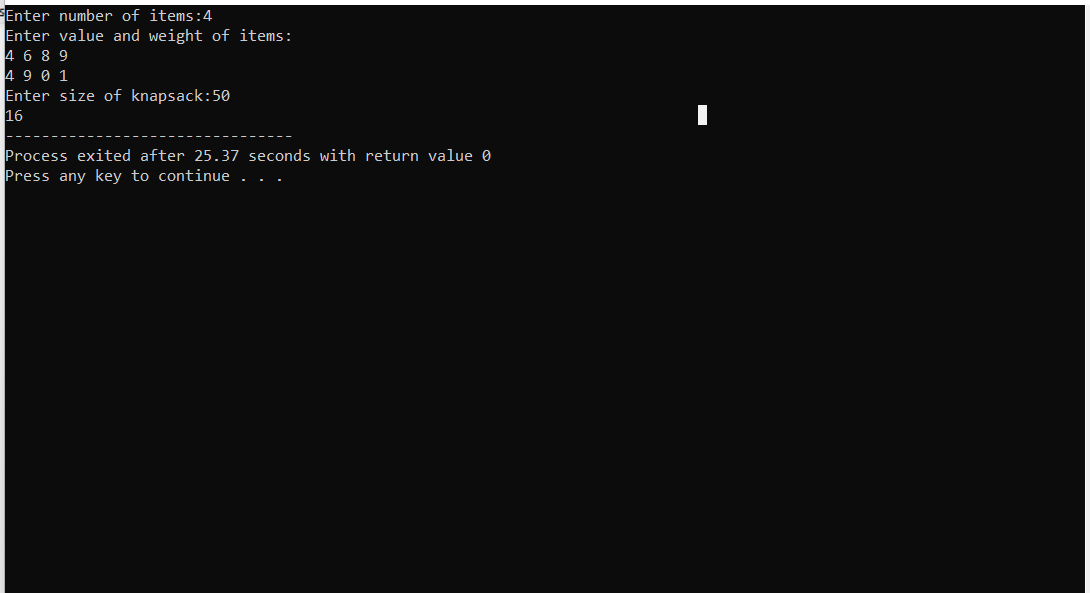
printf("Enter size of knapsack:");

scanf("%d", &W);

printf("%d", knapSack(W, wt, val, n));

return 0;

}



4.minimum spanning tree

#include<stdio.h>

int a,b,u,v,n,i,j,ne=1;

int visited[10]= {

0

}

,min,mincost=0,cost[10][10];

int main()

{

printf("\n Enter the number of nodes:");

scanf("%d",&n);

printf("\n Enter the adjacency matrix:\n");

for (i=1;i<=n;i++)

for (j=1;j<=n;j++) {

scanf("%d",&cost[i][j]);

if(cost[i][j]==0)

cost[i][j]=999;

}

visited[1]=1;

printf("\n");

while(ne<n) {

for (i=1,min=999;i<=n;i++)

for (j=1;j<=n;j++)

if(cost[i][j]<min)

if(visited[i]!=0) {

min=cost[i][j];

a=u=i;

b=v=j;

}

if(visited[u]==0 || visited[v]==0) {

printf("\n Edge %d:(%d %d) cost:%d",ne++,a,b,min);

mincost+=min;

visited[b]=1;

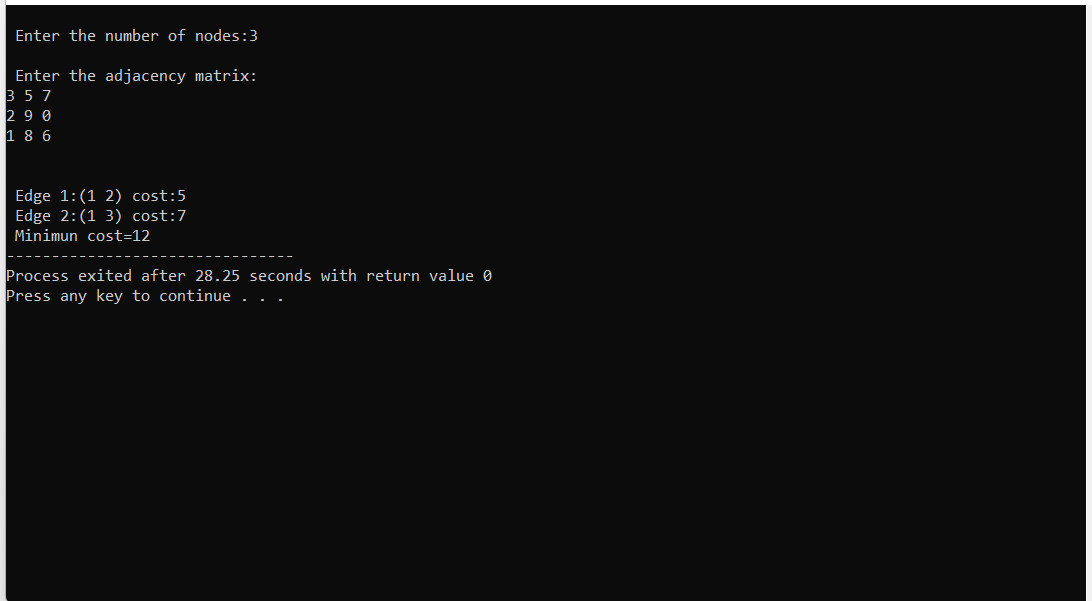
}

cost[a][b]=cost[b][a]=999;

}

printf("\n Minimun cost=%d",mincost);

}



5. binary search

#include <stdio.h>

int binarySearch(int array[], int x, int low, int high) {

if (high >= low) {

int mid = low + (high - low) / 2;

// If found at mid, then return it

if (array[mid] == x)

return mid;

// Search the left half

if (array[mid] > x)

return binarySearch(array, x, low, mid - 1);

// Search the right half

return binarySearch(array, x, mid + 1, high);

}

return -1;

}

int main(void) {

int array[]={3,8,6,4,9,4,7} ;

int n = sizeof(array) / sizeof(array[0]);

int x = 4;

int result = binarySearch(array, x, 0, n - 1);

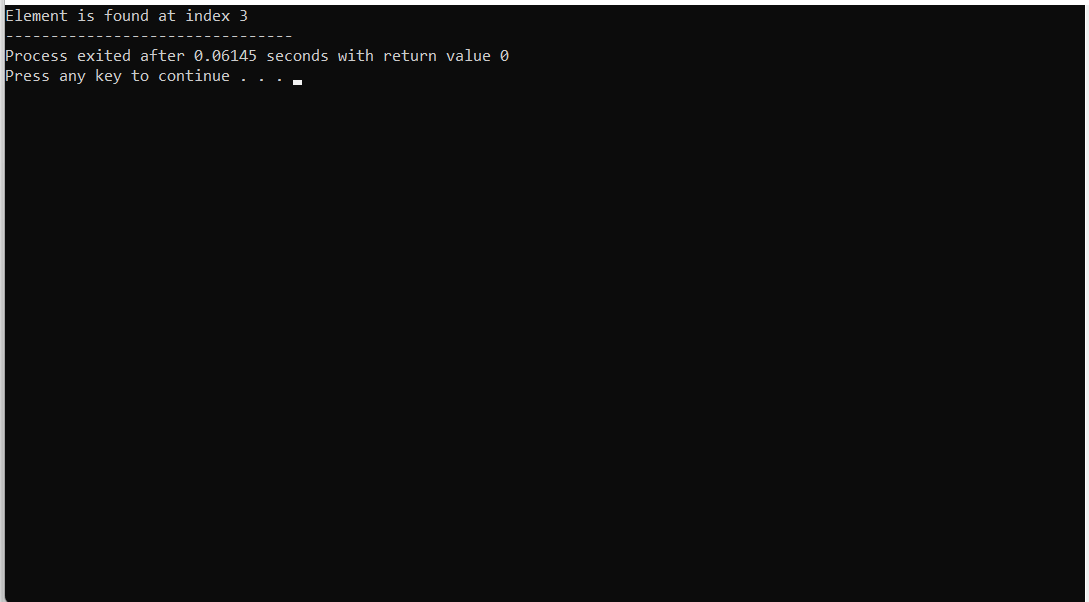
if (result == -1)

printf("Not found");

else

printf("Element is found at index %d", result);

}



6.max and min problem

#include <stdio.h>

#include <conio.h>

int main()

{

int a[1000],i,n,min,max;

printf("Enter size of the array : ");

scanf("%d",&n);

printf("Enter elements in array : ");

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

{

scanf("%d",&a[i]);

}

min=max=a[0];

for(i=1; i<n; i++)

{

if(min>a[i])

min=a[i];

if(max<a[i])

max=a[i];

}

printf("minimum of array is : %d",min);

printf("\nmaximum of array is : %d",max);

return 0;

}

