CSE 2025 DATA STRUCTURES

PROJECT #2 GRAPH IMPLEMENTATION - DIJKSTRA'S ALGORITHM

Hasan Şenyurt – Hasan Mert Yalçın – Melis Çırpan
150120531 150119647 150119669

DATA STRUCTURES PROJECT 2 DIJKSTRA'S ALGORITHM

FUNCTIONS:

SIZE = 50 (DEFINED)

- void read_input (char file[SIZE]):

  char file[SIZE]: char array that represents name of the input file.

  Purpose: Reading an input file which is given by user and taking vertices and length between two vertices to creating arrays to create adjacency matrix.

- void selectionSort (char list[], int list_size):

  char list[]: char array that represents names of vertices. (A,B,C…).
  int list_size: integer that represents size of list of vertices.

  Purpose: Using selection sort algorithm to sort name of vertices as ascendant. For example: {A,D,B,C} -> {A,B,C,D}. It is necessary for printing adjacency matrix.

- void print_matrix (int matrix[SIZE][2], int matrix_size, char vertices[SIZE], int vertex_size, int length[SIZE], int length_size, int choice):

  int matrix[SIZE][2]: integer 2D array that represents indices of names of vertices. For example: {{AB},{BD}} is vertices list. indices of this list are {{01},{13}}.

  int matrix_size: size of matrix.(row)
  char vertices[SIZE]: char array that represents names of vertices. (A,B,C…).
  int vertex_size: integer that represents size of list of vertices.
  int length[SIZE]: integer array that represents list of lengths between two vertices.
  int length_size: integer that represents size of list of lengths.
  int choice: integer that represents choice to print matrix or not. (0 = not, 1 = print)

  Purpose: Creating adjacency matrix and printing it.

- <u>void instructions ():</u>

  <u>Purpose:</u> Printing main menu.

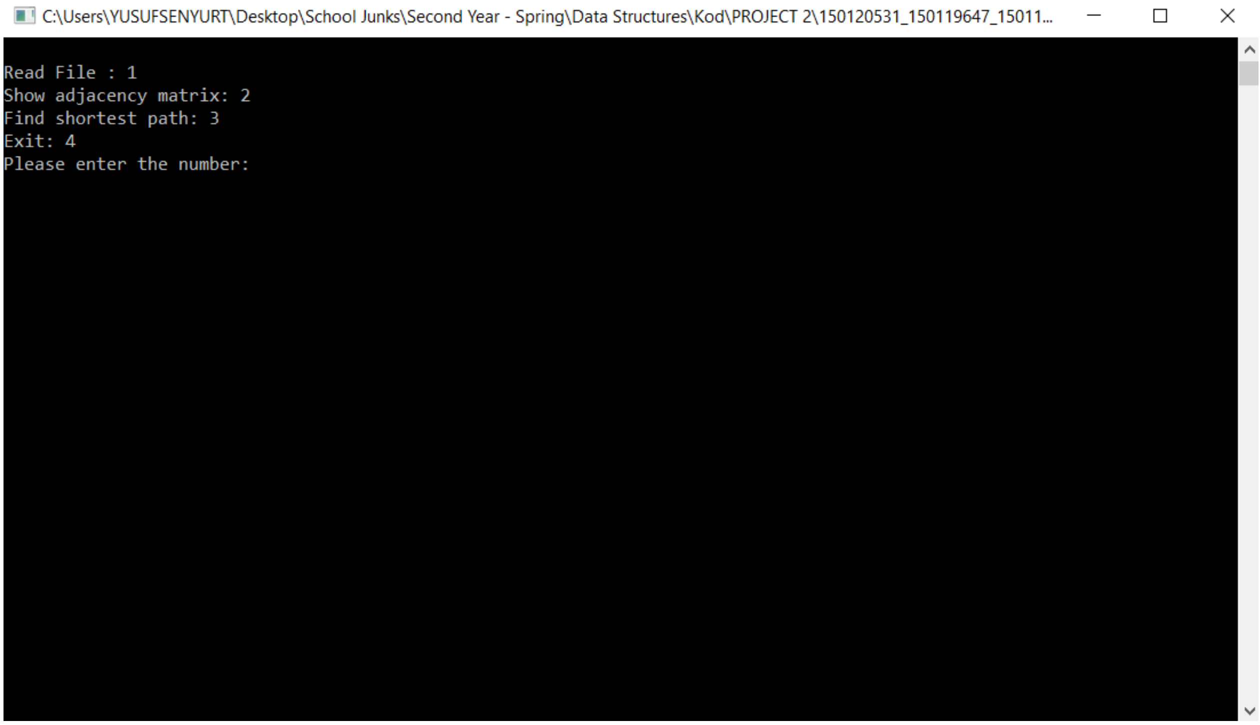
- <u>void dijkstra (int adj_matrix[SIZE][SIZE], int graph_size, int source, int dest):</u>

  int adj_matrix[SIZE][SIZE]: integer 2D array that represents adjacency matrix.
  int graph_size: integer that represents size of adjacency matrix.
  int source: integer that represents index of first vertex (source) that is given by user.
  int dest: integer that represents index of second vertex (destination) that is given by user.

  <u>Purpose:</u> Applying Dijkstra algorithm to find shortest path between two nodes (source and destination).

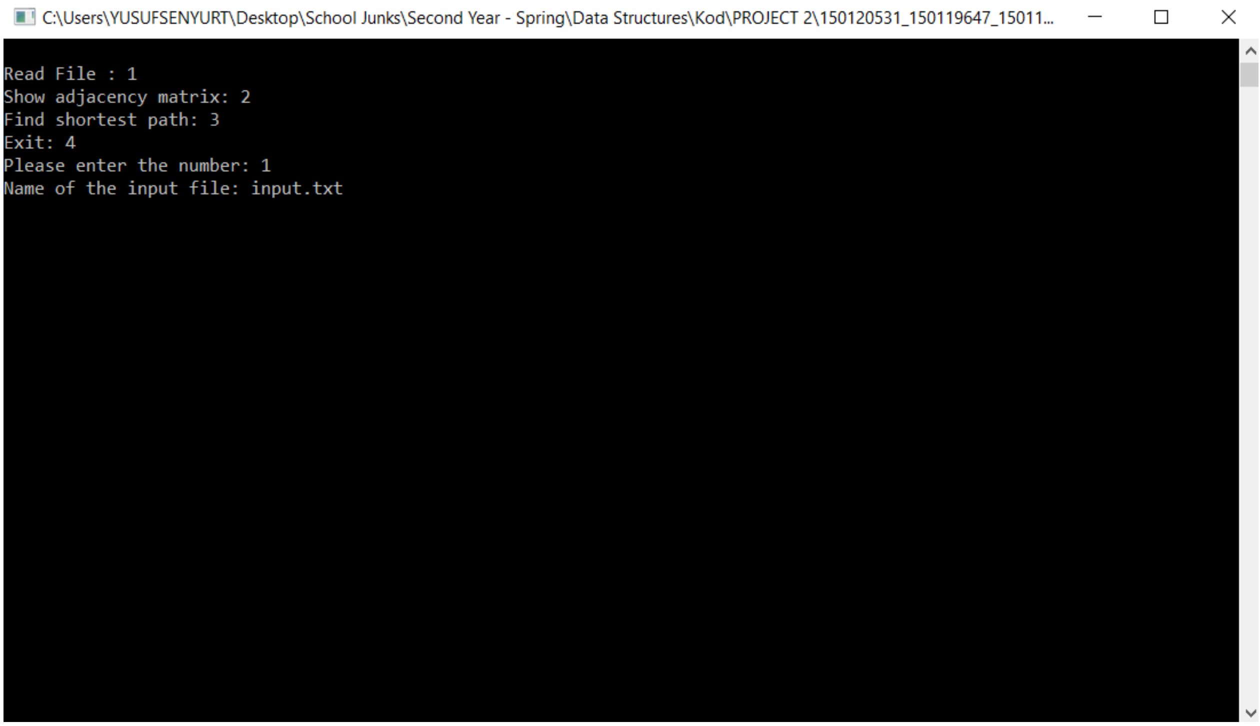
<u>SCREENSHOTS:</u>

1) <u>Main Menu:</u>

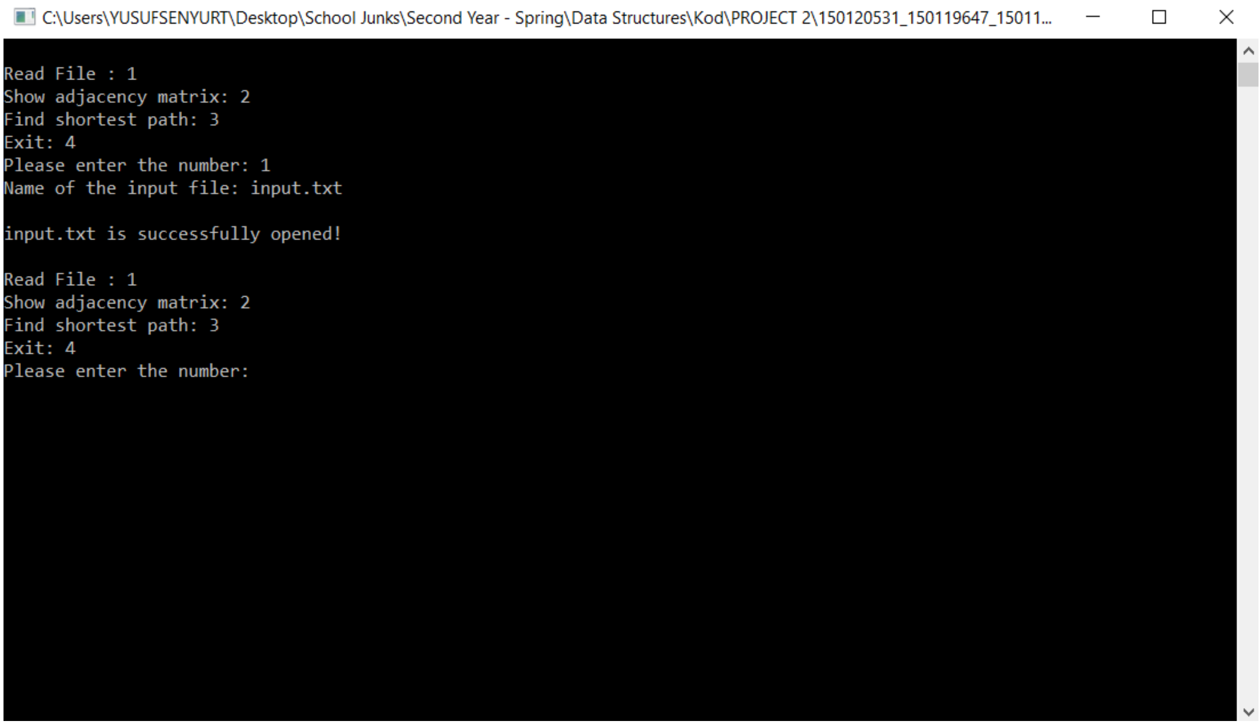

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number:
```

2) Reading File Input:

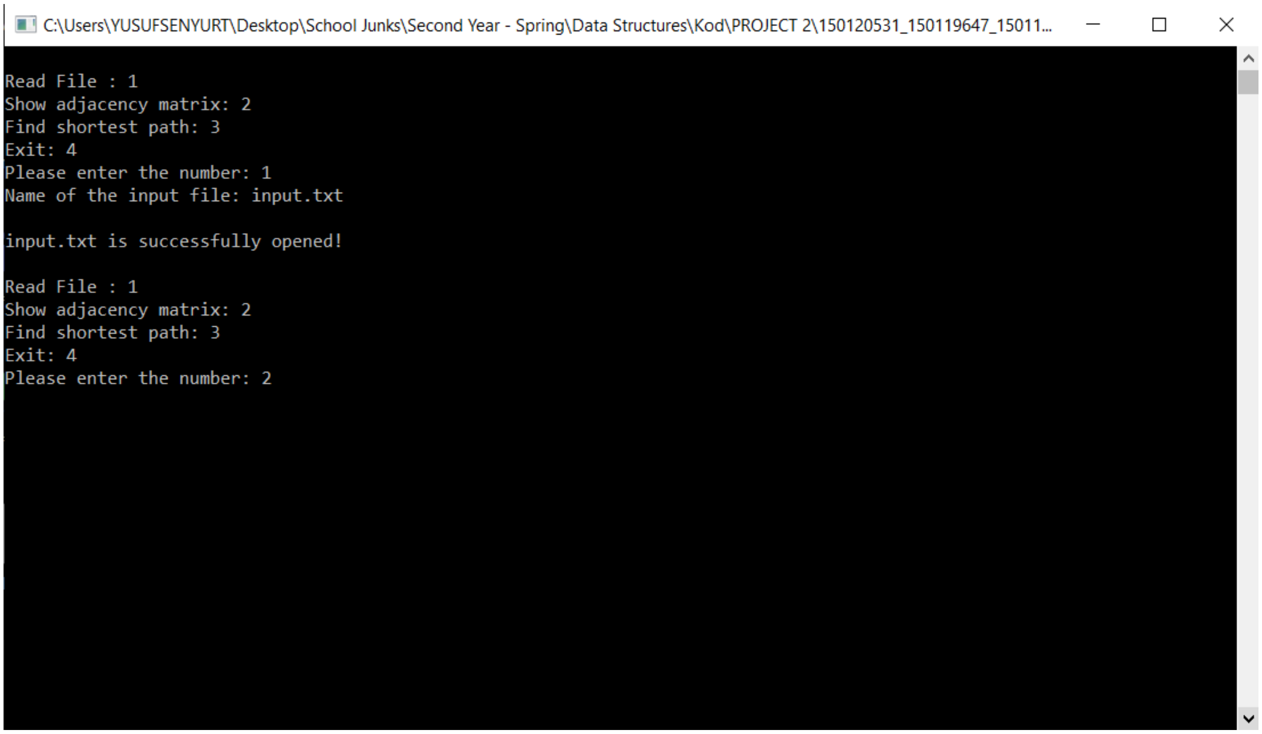

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 1
Name of the input file: input.txt
```

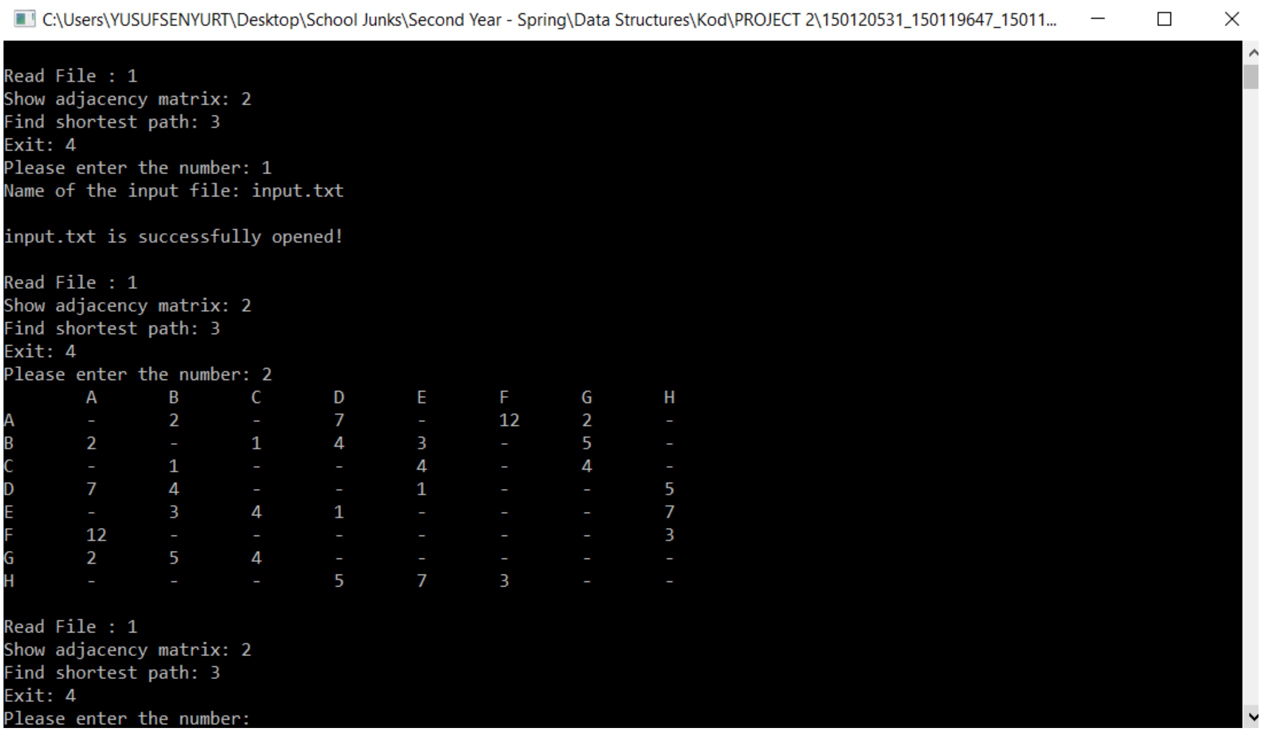

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 1
Name of the input file: input.txt

input.txt is successfully opened!

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number:
```

3) Showing an Adjacency Matrix:

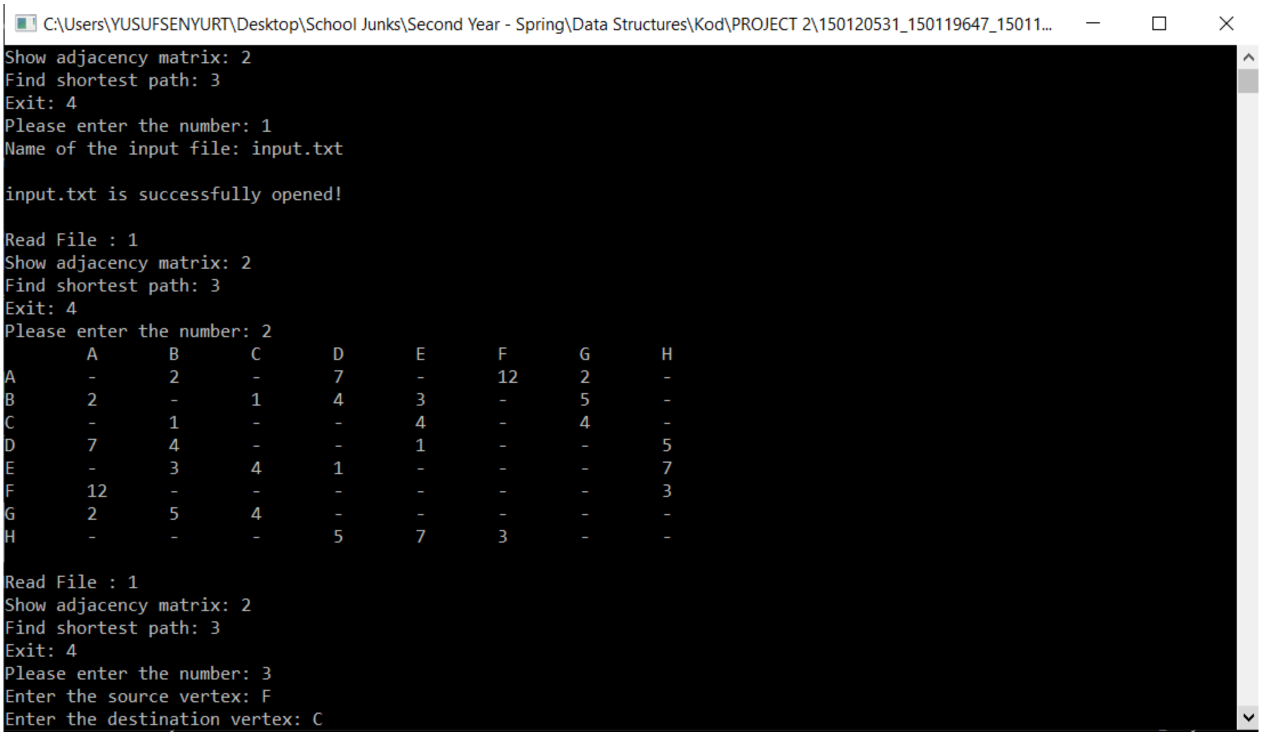

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 1
Name of the input file: input.txt

input.txt is successfully opened!

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 2
```

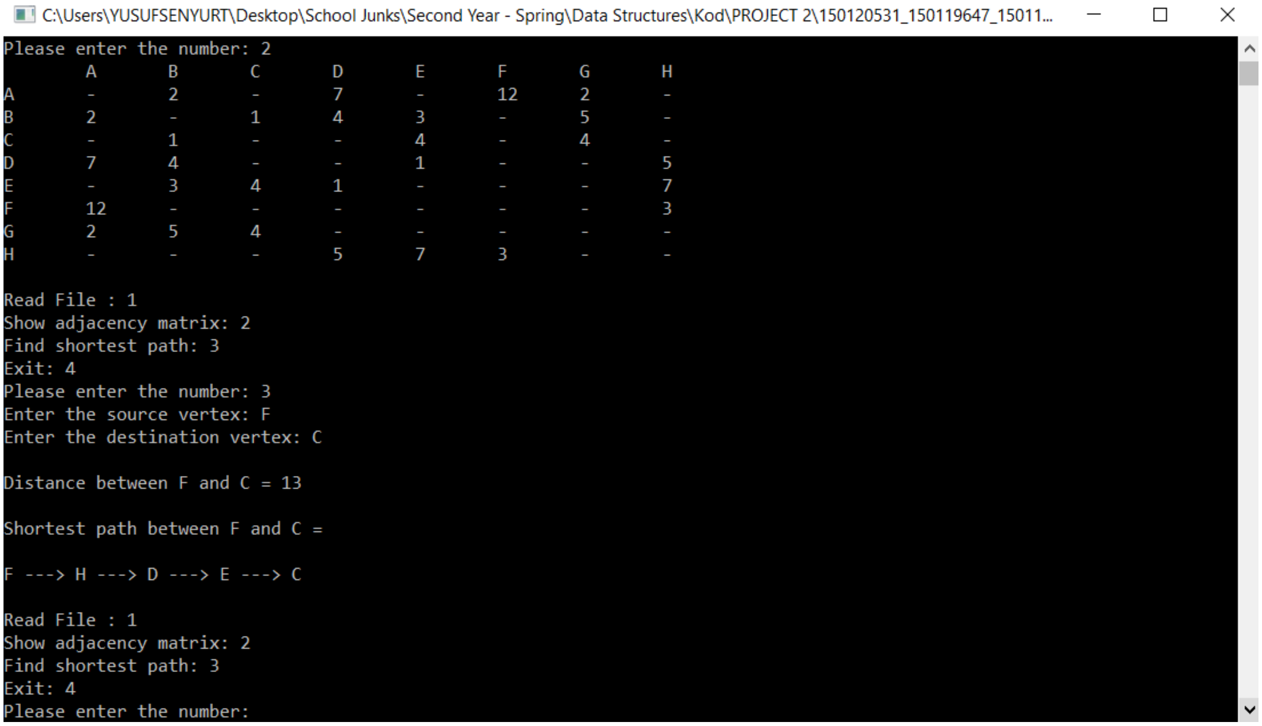

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 1
Name of the input file: input.txt

input.txt is successfully opened!

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 2
        A       B       C       D       E       F       G       H
A       -       2       -       7       -       12      2       -
B       2       -       1       4       3       -       5       -
C       -       1       -       -       4       -       4       -
D       7       4       -       -       1       -       -       5
E       -       3       4       1       -       -       -       7
F       12      -       -       -       -       -       -       3
G       2       5       4       -       -       -       -       -
H       -       -       -       5       7       3       -       -

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number:
```

4) Finding Shortest Path:

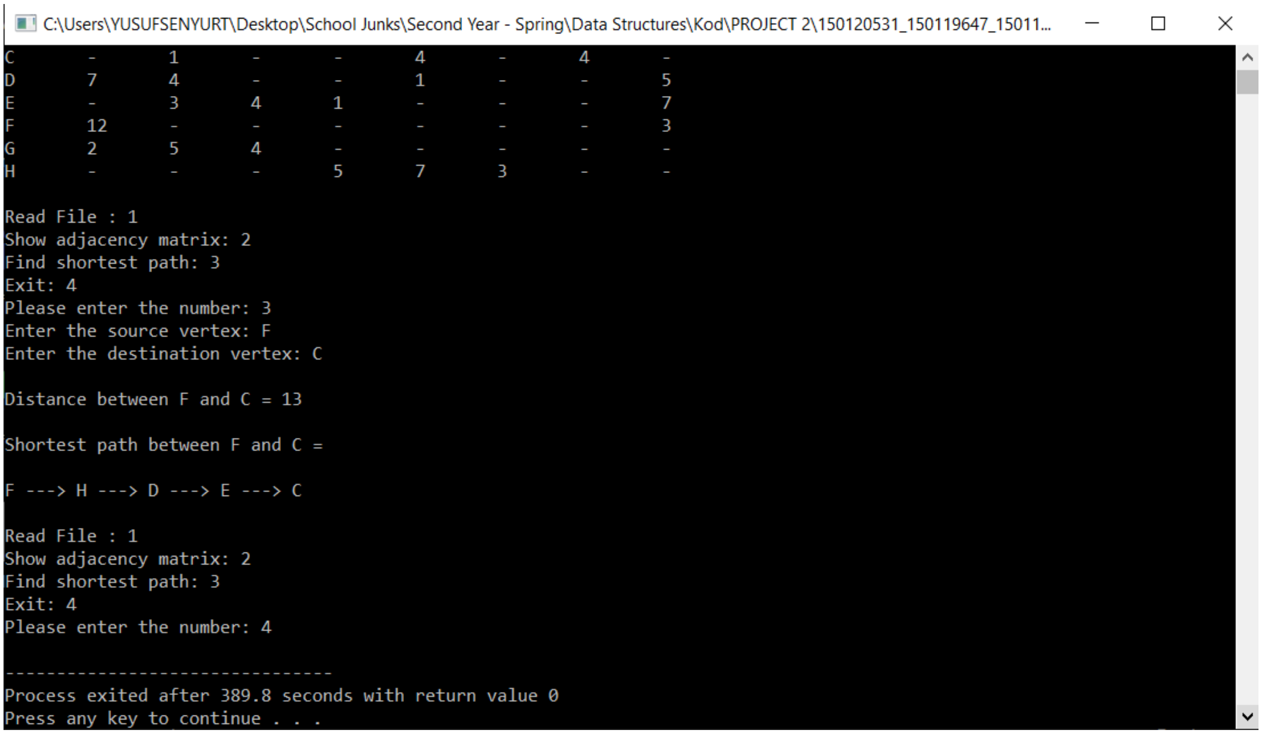

C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 1
Name of the input file: input.txt

input.txt is successfully opened!

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 2
        A       B       C       D       E       F       G       H
A       -       2       -       7       -       12      2       -
B       2       -       1       4       3       -       5       -
C       -       1       -       -       4       -       4       -
D       7       4       -       -       1       -       -       5
E       -       3       4       1       -       -       -       7
F       12      -       -       -       -       -       -       3
G       2       5       4       -       -       -       -       -
H       -       -       -       5       7       3       -       -

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 3
Enter the source vertex: F
Enter the destination vertex: C
```


C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
Please enter the number: 2
        A       B       C       D       E       F       G       H
A       -       2       -       7       -       12      2       -
B       2       -       1       4       3       -       5       -
C       -       1       -       -       4       -       4       -
D       7       4       -       -       1       -       -       5
E       -       3       4       1       -       -       -       7
F       12      -       -       -       -       -       -       3
G       2       5       4       -       -       -       -       -
H       -       -       -       5       7       3       -       -

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 3
Enter the source vertex: F
Enter the destination vertex: C

Distance between F and C = 13

Shortest path between F and C =

F ---> H ---> D ---> E ---> C

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number:
```

5) Exit:


C:\Users\YUSUFSENYURT\Desktop\School Junks\Second Year - Spring\Data Structures\Kod\PROJECT 2\150120531_150119647_15011...

```
C       -       1       -       -       4       -       4       -
D       7       4       -       -       1       -       -       5
E       -       3       4       1       -       -       -       7
F       12      -       -       -       -       -       -       3
G       2       5       4       -       -       -       -       -
H       -       -       -       5       7       3       -       -

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 3
Enter the source vertex: F
Enter the destination vertex: C

Distance between F and C = 13

Shortest path between F and C =

F ---> H ---> D ---> E ---> C

Read File : 1
Show adjacency matrix: 2
Find shortest path: 3
Exit: 4
Please enter the number: 4

--------------------------------
Process exited after 389.8 seconds with return value 0
Press any key to continue . . .
```

COMPLETED AND INCOMPLETE PARTS OF PROJECT:

- Reading a file input from user is completed.
- Creating and printing adjacency matrix is completed.
- Finding shortest distance between two vertices and finding path of it by using Dijkstra’s algorithm are completed.
- Main menu functionality is working. It is completed.