All_pair_Shortest_path

Write Java program to Implement All-Pairs Shortest Paths problem using Floyd's algorithm

Input Format

5 0 5 999 2 999 999 0 2 999 999 3 999 0 999 7 999 999 4 0 1 1 3 999 999 0

Constraints

no constraints

Output Format

all pair shortest paths matrix. 0 5 6 2 3 5 0 2 7 8 3 8 0 5 6 2 4 4 0 1 1 3 5 3 0

Sample Input 0

```
5
0 5 999 2 999
999 0 2 999 999
3 999 0 999 7
999 999 4 0 1
1 3 999 999 0
```

Sample Output 0

```
all pair shortest paths matrix.
0 5 6 2 3
5 0 2 7 8
3 8 0 5 6
2 4 4 0 1
1 3 5 3 0
```


Java 7

```
1  //224G1A0553
2  import java.util.*;
3  public class Floyds {
4  static int n,i,j,k;
5  public void floyd(int n , int[][] cost) {
6  for(k=1;k<=n;k++) {
7  for(i=1;i<=n;i++) {
8  for(j=1;j<=n;j++) {
9  cost[i][j]=min(cost[i][j],cost[i][k]+cost[k][j]);
10 }
11 }
12 }
```

```
13 System.out.println("all pair shortest paths matrix.");
14 for(i=1;i<=n;i++) {
15 for(j=1;j<=n;j++) {
16 System.out.print(cost[i][j]+" ");
17 }
18 System.out.println();
19 }
20 }
21 public int min(int i,int j) {
22 if(i<j)
23 return i;
24 else
25 return j;
26 }
27 public static void main(String[] args) {
28 Scanner sc=new Scanner(System.in);
29 //System.out.println("Enter the no of vertices: ");
30 n=sc.nextInt();
31 int cost[][]=new int[n+1][n+1];
32 //System.out.println("Enter the cost matrix:");
33 for(i=1;i<=n;i++)
34 for(j=1;j<=n;j++)
35 cost[i][j]=sc.nextInt();
36 Floyds f = new Floyds();
37 f.floyd(n,cost);
38 }
39 }
```

Line: 39 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
5
0 5 999 2 999
999 0 2 999 999
3 999 0 999 7
999 999 4 0 1
1 3 999 999 0
```

Your Output (stdout)

```
all pair shortest paths matrix.
0 5 6 2 3
5 0 2 7 8
3 8 0 5 6
2 4 4 0 1
1 3 5 3 0
```

Expected Output

```
all pair shortest paths matrix.
0 5 6 2 3
5 0 2 7 8
3 8 0 5 6
```

```
2 4 4 0 1
1 3 5 3 0
```