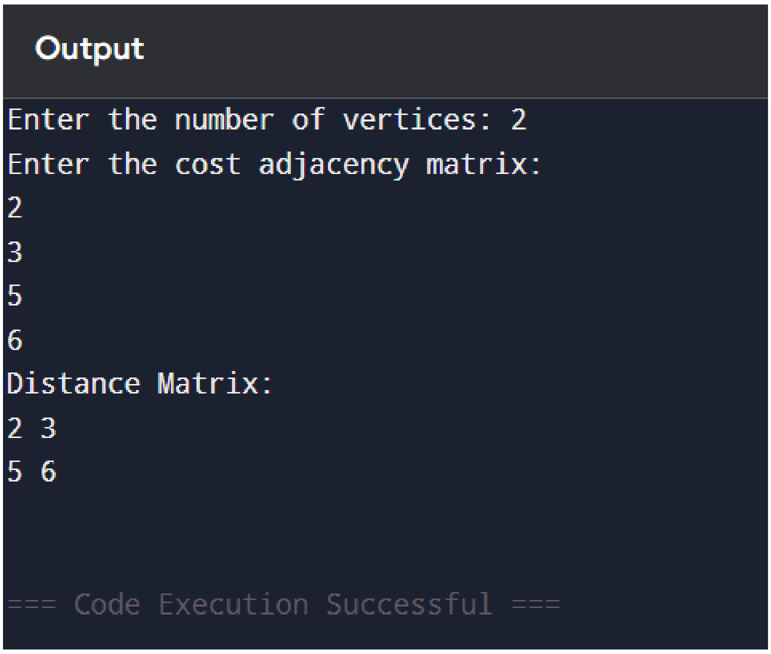
Implement All Pair Shortest paths problem using Floyd’s algorithm.

```
#include <stdio.h>

int a[10][10], D[10][10], n;

void floyd(int [][10], int);
int min(int, int);

int main() {
   int i, j;

   printf("Enter the number of vertices: ");
   scanf("%d", &n);

   printf("Enter the cost adjacency matrix:\n");
   for (i = 0; i < n; i++) {
      for (j = 0; j < n; j++) {
         scanf("%d", &a[i][j]);
      }
   }

   floyd(a, n);

   printf("Distance Matrix:\n");
   for (i = 0; i < n; i++) {
      for (j = 0; j < n; j++) {
         printf("%d ", D[i][j]);
      }
```

```
        printf("\n");
    }

    return 0;
}

void floyd(int a[][10], int n) {
    int i, j, k;

    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            D[i][j] = a[i][j];
        }
    }

    for (k = 0; k < n; k++) {
        for (i = 0; i < n; i++) {
            for (j = 0; j < n; j++) {
                D[i][j] = min(D[i][j], D[i][k] + D[k][j]);
            }
        }
    }
}

int min(int a, int b) {
    return (a < b) ? a : b;
}
```



```
Enter the number of vertices: 2
Enter the cost adjacency matrix:
2
3
5
6
Distance Matrix:
2 3
5 6


=== Code Execution Successful ===
```