

```
public static void Print(int[][] arr) {
                    // TODO Auto-generated method stub
                    int minc = 0, minr = 0, maxc = arr[0].length - 1, maxr = arr.length - 1;
                    int te = arr.length * arr[0].length;
                    int c = 0;
                    while (c < te)
                     for (int i = minc; i <= maxc; i++) {</pre>
                          System.out.print(arr[minr][i] + " ");
                          C++;
            19
                       minr++;
                       for (int i = minr; i <= maxr; i++) {</pre>
   (′9 )
6
        16
            22
                          System.out.print(arr[i][maxc] + " ");
                          C++;
13
            24
   14
                       maxc--;
21
   23
        26
            30
                       (for (int i = maxc; i >= minc; i--) {
                          System.out.print(arr[maxr][i] + " ");
                          C++;
                       maxr--;
                       for (int i = maxr; i >= minr; i--) {
                          System.out.print(arr[i][minc] + " ");
                          C++;
                       minc++;
```

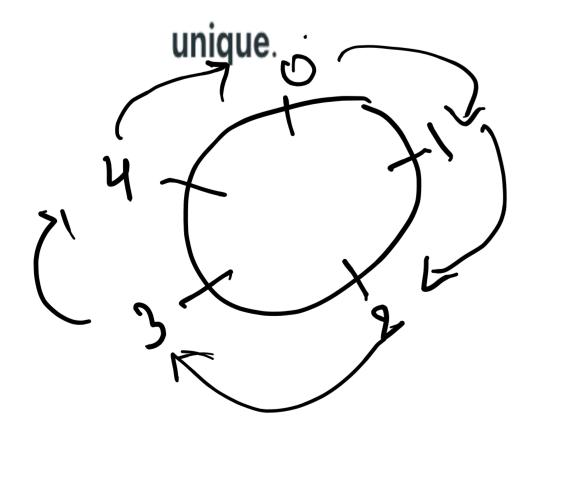
```
public static void main(String[] args) {
     // TODO Auto-generated method stub
     // TODO Auto-generated method stub
     int[][] arr = { { 1, 2, 3, 4 },
                   \{5, 6, 7, 8\}, \{9, 10, 11, 12\}\};
     Print(arr);
  public static void Print(int[][] arr) {
     // TODO Auto-generated method stub
     int minc = 0, minr = 0, maxc = arr[0].length - 1, maxr = arr.length - 1;
     int te = arr.length * arr[0].length;
     int c = 0;
     while (c < te) { \o∠\\
                                                                 2 SIMAXC
       for (int i = minc; i <= maxc; i++) {</pre>
          System.out.print(arr[minr][i] + " ");
           C++;
                                                            5
        for (int i = minr; i <= maxr; i++) {
           System.out.print(arr[i][maxc] + " ");
           C++;
                                                       472 +3 +1 =10+2=12
        maxc---
        for (int i = \max_{i \in I} i > = \min_{i \in I} i - i) {
          System.out.print(arr[maxr][i] + " ");
           C++;
        maxr--;
        for (int i = maxr; i >= minr; i--) {
           System.out.print(arr[i][minc] + " ");
           C++;
        minc++;
```

There are n gas stations along a circular route, where the amount of gas at the i^{th} station is gas [i].

gass (1) - (m' () (0

You have a car with an unlimited gas tank and it costs cost[i] of gas to travel from the i^{th} station to its next $(i + 1)^{th}$ station. You begin the journey with an empty tank at one of the gas stations.

Given two integer arrays gas and cost, return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1. If there exists a solution, it is **guaranteed** to be



```
public static int canComplete(int[] gas, int[] cost) {
    int total = 0;
    for (int i = 0; i < cost.length; i++) {
        total += gas[i] - cost[i];
    }
    if (total < 0) {
        return -1;
    }
    int idx = 0; i < cost.length; i++) {
        curr = 0;
        for (int i = 0; i < cost.length; i++) {
            curr += gas[i] - cost[i];
        idx = i + 1;
        }
    }
}</pre>
```