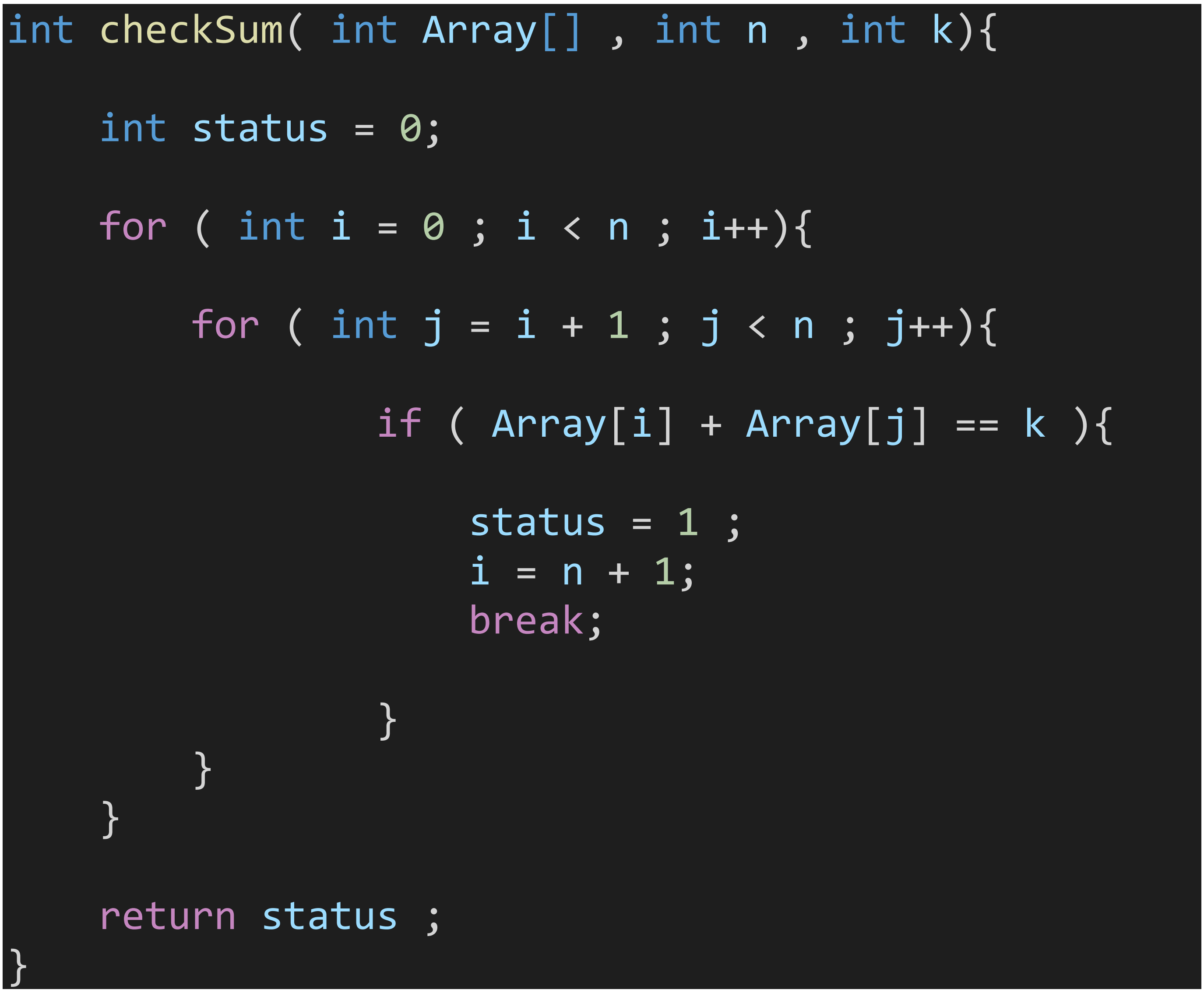
۱

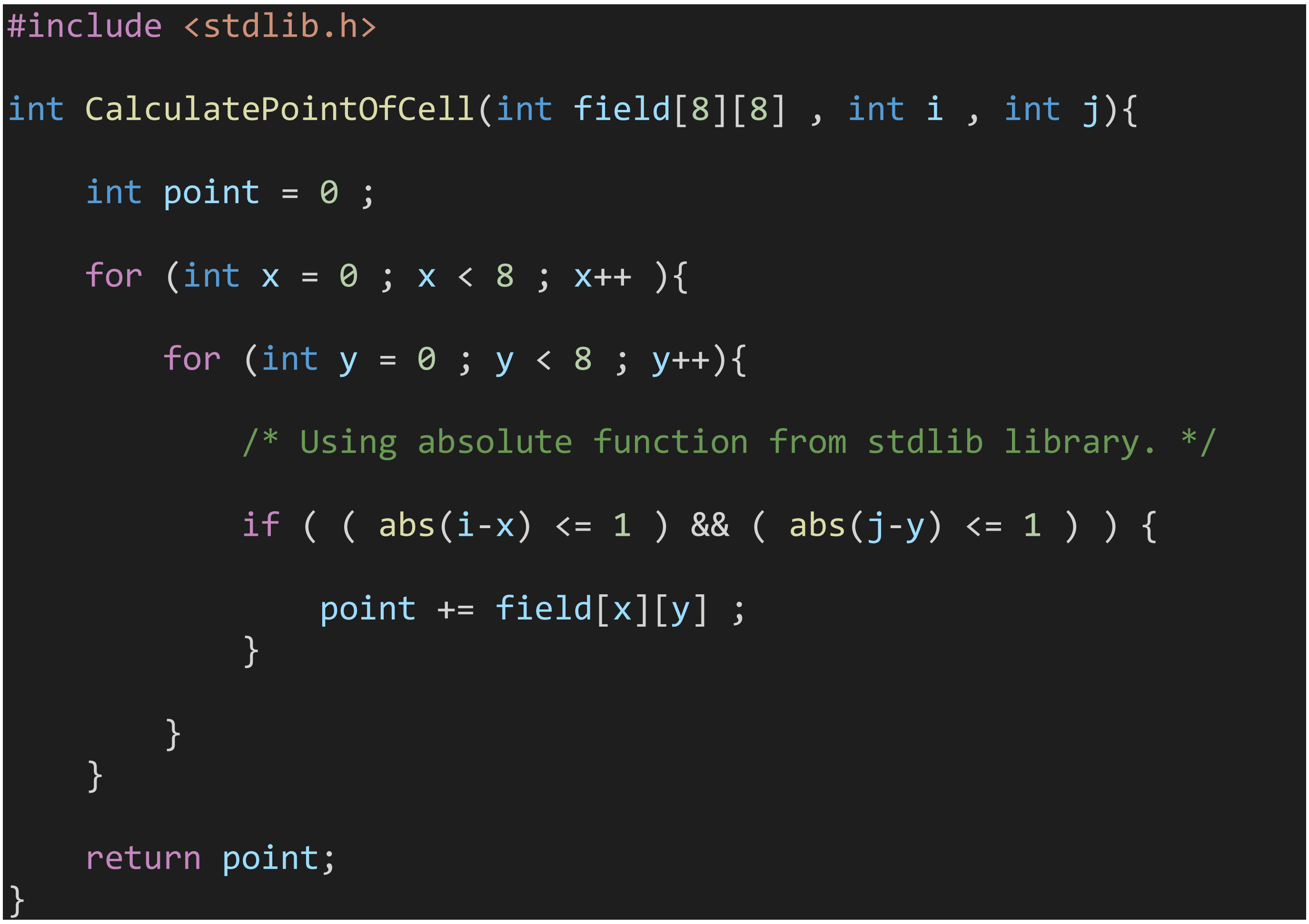
الف)

```
1 ) Start
2 ) i=0 , j=0 , k=0 , status = 0
3 ) Get k from user
4 ) While i < n
          j = i + 1
          While j < n
              If ( Array[i] + Array[j] == k ) then
                  Print("true")
                  status = 1
                  j = n + 1
                  i = n + 1
              Endif
              j = j + 1


            i = i + 1
5 ) if ( status == 0 )  then
        Print("false")
    Endif
6 ) End
```

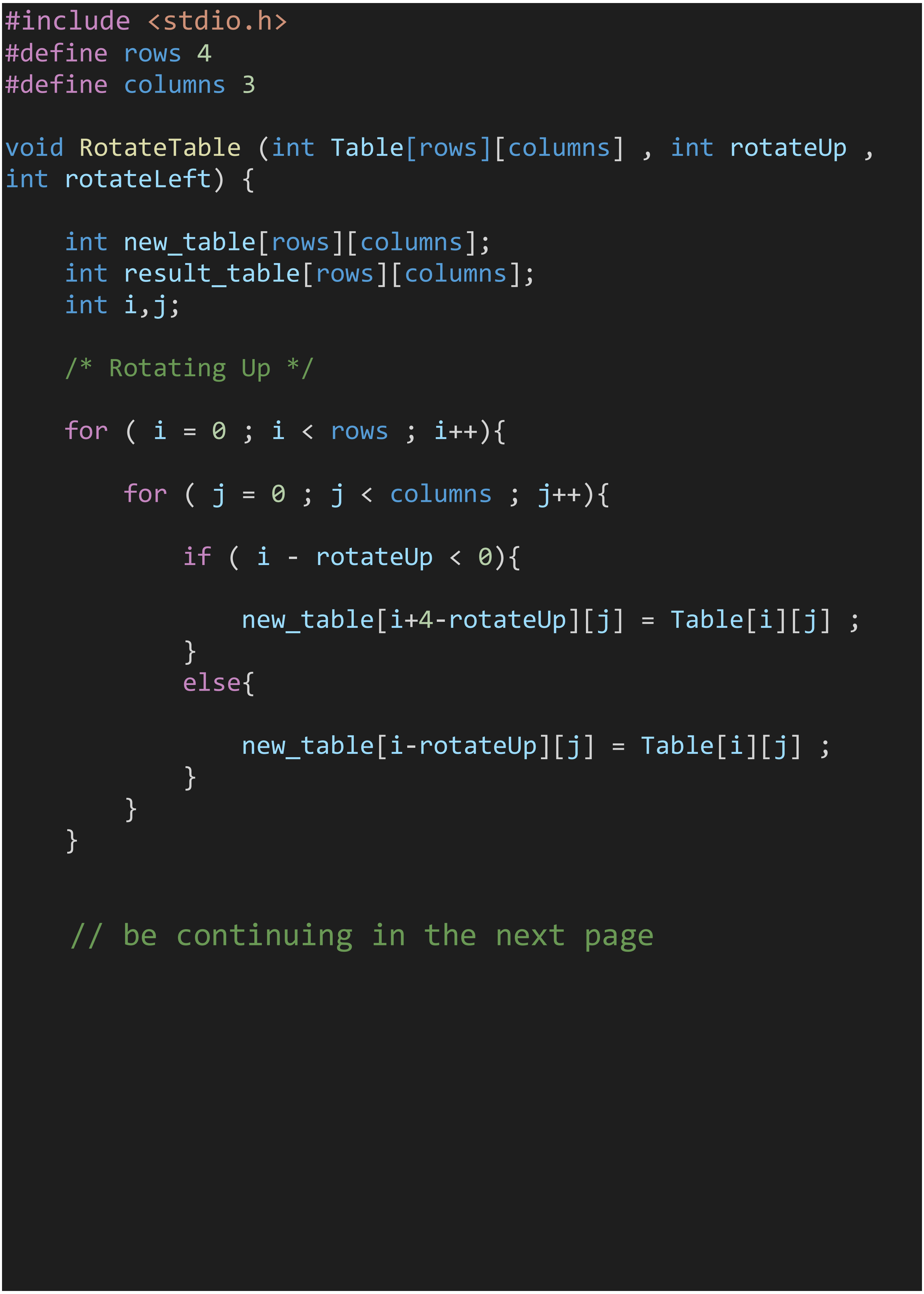
ب)



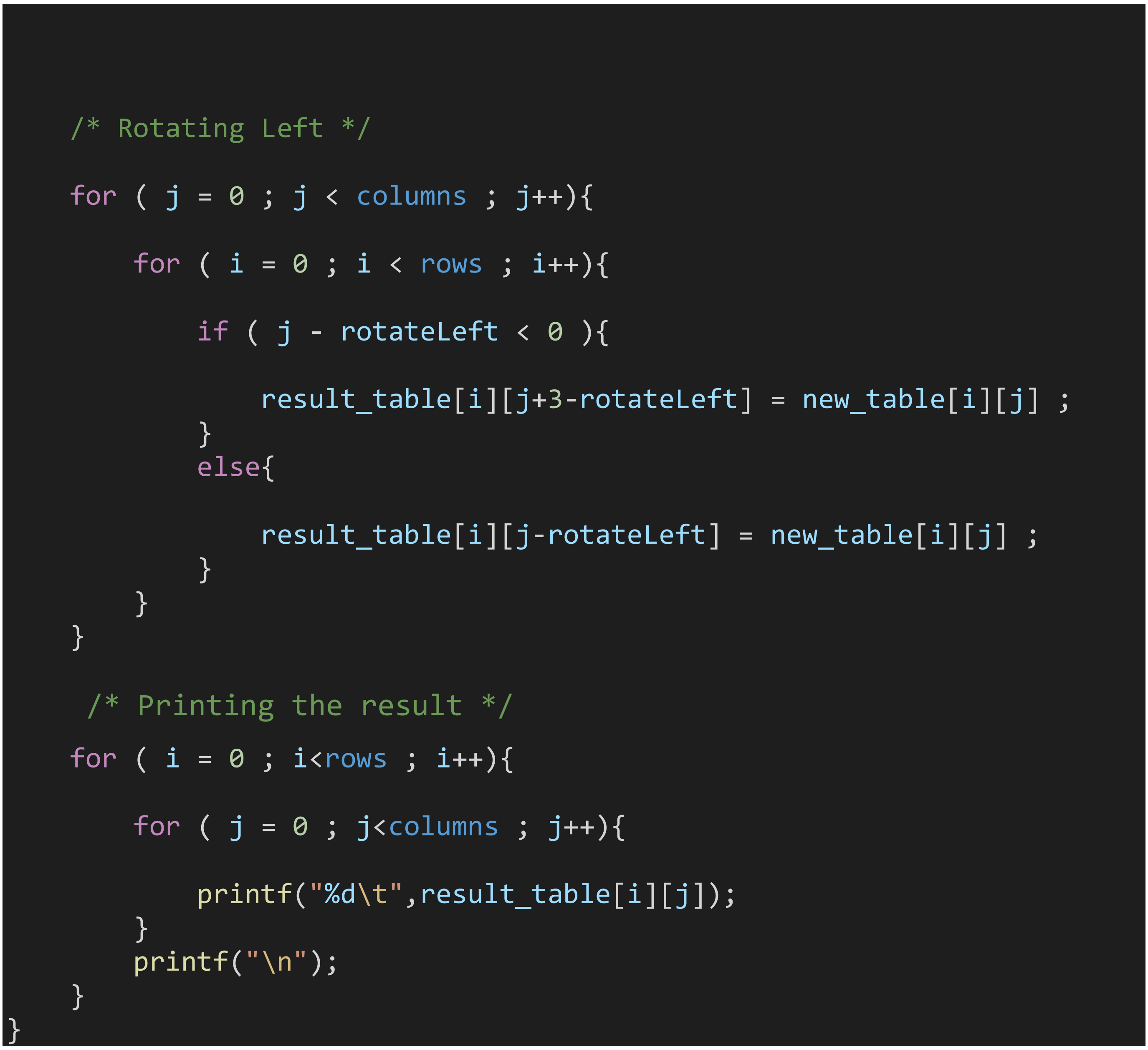
```
int checkSum( int Array[] , int n , int k){

    int status = 0;

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

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

                if ( Array[i] + Array[j] == k ){

                    status = 1 ;
                    i = n + 1;
                    break;

                }
        }
    }

    return status ;
}
```



```c
#include <stdlib.h>

int CalculatePointOfCell(int field[8][8] , int i , int j){

    int point = 0 ;

    for (int x = 0 ; x < 8 ; x++ ){

        for (int y = 0 ; y < 8 ; y++){

            /* Using absolute function from stdlib library. */

            if ( ( abs(i-x) <= 1 ) && ( abs(j-y) <= 1 ) ) {

                point += field[x][y] ;
            }

        }
    }

    return point;
}
```



```
#include <stdio.h>
#define rows 4
#define columns 3

void RotateTable (int Table[rows][columns] , int rotateUp ,
int rotateLeft) {

    int new_table[rows][columns];
    int result_table[rows][columns];
    int i,j;

    /* Rotating Up */

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

        for ( j = 0 ; j < columns ; j++){

            if ( i - rotateUp < 0){

                new_table[i+4-rotateUp][j] = Table[i][j] ;
            }
            else{

                new_table[i-rotateUp][j] = Table[i][j] ;
            }
        }
    }


    // be continuing in the next page
```



```
  /* Rotating Left */

  for ( j = 0 ; j < columns ; j++){

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

      if ( j - rotateLeft < 0 ){

        result_table[i][j+3-rotateLeft] = new_table[i][j] ;
      }
      else{

        result_table[i][j-rotateLeft] = new_table[i][j] ;
      }
    }
  }

   /* Printing the result */

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

    for ( j = 0 ; j<columns ; j++){

      printf("%d\t",result_table[i][j]);
    }
    printf("\n");
  }
}
```

۴

```
1 ) Start
2 ) i = 1 , j = 0 , Max_num = 0 , Repeats = 0
3 ) Get Array[n] from user
4 ) Max_num = Array[0]
5 ) While i < n
            If ( Array[i] > Max_num ) then
                  Max_num = Array[i]
            Endif
            i = i + 1

6 ) While  j < n
             If ( Array[j] == Max_num ) then
                  Repeats = Repeats + 1
             Endif
             j = j + 1

7 ) print ( Repeats )
8 ) End
```

```
1 ) Start
2 ) L = 0  , count = 0 , val = 0
3 ) Get str from user
4 ) CalculateValue(str) :        // This is a function.
        L = length_of_str_without_counting_null
        If ( L == 2 ) then
            return 1
        Elseif (   str[0]  == '('    and   str[L-1] == ')'   ) then
            count = 0 , i = 1
            while ( i <= L – 2 )
                if (   str[i] == '('   ) then
                    count ++
                Endif
                i = i + 1
            Endwhile
            return ( 2 to the power of count )
```