Tutorial 3 – Arrays – Suggested Answers

1. (histogram)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void getFrequency(int histogram[10], int n);
void printFrequency(int histogram[10]);
int main()
{
  int frequencies[10];
  int total;

  printf("Please input the number of random numbers: ");
  scanf("%d", &total);
  srand(time(NULL));
  getFrequency(frequencies, total);
  printFrequency(frequencies);
  return 0;
}
void getFrequency(int histogram[10], int n)
{
  int count;
  // int category;

  for (count = 0; count < 10; count++)
    histogram[count] = 0;
  for (count = 0; count < n; count++)
    histogram[(rand() % 100)/10]++;
    /* category = rand() % 100)/10;
      histogram[category]++; */
}
void printFrequency(int histogram[10])
{
  int count, index;

  for (count = 0; count < 10; count++) {
    printf("%2d--%2d  |", count*10, count*10+9);
    for (index = 0; index < histogram[count]; index++ )
      putchar('*');
    putchar('\n');
  }
}
```

2. (transpose2D)

```
#include <stdio.h>
#define SIZE 10
void transpose2D(int ar[][SIZE], int rowSize, int colSize);
void display(int ar[][SIZE], int rowSize, int colSize);
```

```c
int main()
{
  int ar[SIZE][SIZE], rowSize, colSize;
  int i,j;

  printf("Enter row size of the 2D array: \n");
  scanf("%d", &rowSize);
  printf("Enter column size of the 2D array: \n");
  scanf("%d", &colSize);
  printf("Enter the matrix (%dx%d): \n", rowSize, colSize);
  for (i=0; i<rowSize; i++)
    for (j=0; j<colSize; j++)
      scanf("%d", &ar[i][j]);
  printf("transpose2D(): \n");
  transpose2D(ar, rowSize, colSize);
  display(ar, rowSize, colSize);
  return 0;
}
void display(int ar[][SIZE], int rowSize, int colSize)
{
  int l,m;
  for (l = 0; l < rowSize; l++) {
    for (m = 0; m < colSize; m++)
      printf("%d ", ar[l][m]);
    printf("\n");
  }
}
void transpose2D(int ar[][SIZE], int rowSize, int colSize)
{
  int h, k;
  int temp;

  for (h = 1; h < rowSize; h++)
    for (k = 0; k < h; k++) {
      temp = ar[h][k];
      ar[h][k] = ar[k][h];
      ar[k][h] = temp;
    }
}
```

3. (reduceMatrix2D)

```c
#include <stdio.h>
#define SIZE 10
void reduceMatrix2D(int ar[][SIZE], int rowSize, int colSize);
void display(int ar[][SIZE], int rowSize, int colSize);
int main()
{
  int ar[SIZE][SIZE], rowSize, colSize;
  int i,j;
  printf("Enter row size of the 2D array: \n");
```

```
   scanf("%d", &rowSize);
   printf("Enter column size of the 2D array: \n");
   scanf("%d", &colSize);
   printf("Enter the matrix (%dx%d): \n", rowSize, colSize);
   for (i=0; i<rowSize; i++)
     for (j=0; j<colSize; j++)
       scanf("%d", &ar[i][j]);
   reduceMatrix2D(ar, rowSize, colSize);
   printf("reduceMatrix2D(): \n");
   display(ar, rowSize, colSize);
   return 0;
 }
 void display(int ar[][SIZE], int rowSize, int colSize)
 {
   int l,m;
   for (l = 0; l < rowSize; l++) {
     for (m = 0; m < colSize; m++)
       printf("%d ", ar[l][m]);
     printf("\n");
   }
 }
 void reduceMatrix2D(int ar[][SIZE], int rowSize, int colSize)
 {
   int i, j, sum; // i for row, j for column
   /* for each column */
   for (j = 0; j < colSize; j++){
     sum = 0;
     // process the row below matrix[j][j] of the column
     for (i = j+1; i < rowSize; i++){
       sum += ar[i][j];
       ar[i][j] = 0;
     }
     ar[j][j] += sum;
   }
 }
```

4. Suggested Answer

(a) The function add1() has two parameters. The first one is an array address and the second one is the size of the array. So the function adds 1 to every element of the one dimensional array. When the function is called in the for statement at line a by

```
add1(array[h], 4);
```

array[h] is an one dimensional array of 4 integers. It is the (h+1)th row of the two dimensional array 'array'. In fact, array[h] is the address of the first element of the (h+1)th row. So every function call works on one row of the two dimensional array.

Output:

```
 2       3       4       5
 6       7       8       9
10      11      12      13
```

(b) When the for statement at line a is replaced by add1(array[0], 3*4), it is passing the address of the first element of the first row to add1() and telling the function that the array size is 12. So add1() works on an one dimensional array starting at array[0] and with 12 elements.

<u>Output:</u>

```
 2       3       4       5
 6       7       8       9
10      11      12      13
```