CS100 Tutorial 3 (Fall semester, 2018)

Solutions

Q1.

Suggested code:

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

void main(void)
{
    int frequencies[10];
    int total;

    printf("Please input the number of random numbers: ");
    scanf("%d", &total);
    srand(time(NULL));
    getFrequency(frequencies, total);
    printFrequency(frequencies);
}

void getFrequency(int histogram[10], int n)
{
    int count;

    for (count = 0; count < 10; count++)
        histogram[count] = 0;
    for (count = 0; count < n; count++)
        histogram[(rand() % 100)/10]++;
}

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');
    }
}
```

Q2.

Suggested answer:

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.

When the for statement at line a is replace 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. Thus, add1() works on an one-dimensional array starting at array[0] and with 12 elements.

Q3.

Suggested answer:

```
How are ya, sweetie?How are ya, sweetie?
Beat the clock.
eat the clock.
win a toy.
win a snoopy.
win
chat
hat
at
t
t
at
How are ya, sweetie?
```

Q4.

Suggested code:

```
void stringncpy(char *s1, char *s2, int n)
{
     int k, h;

     for (k = 0; k < n; k++)   {
         if (s2[k] != '\0')
             s1[k] = s2[k];
         else
            break;
     }
     for (h = k; h < n; h++)
         s1[h] = '\0';
}
```