CS100 Tutorial 3 (Fall semester, 2018)

1. Write a program that will draw the histogram for n integers from 0 to 99. The value of n is input by the user. Each of the n numbers will be generated by calling rand() % 100. The program will consist of two functions: (1) collect the frequency distribution of the numbers; (2) print the histogram. An example histogram is shown here:

```
0 – 9       |********************
10 – 19     |************
20 – 29     |***************
30 – 39     |**
...
90 – 99     |******************
```

2. Explain how the addition of 1 to every element of the 2-dimensional array "Ar" is done in the following program:

```
#include <stdio.h>

void add1(int ar[], int size);
void display(int ar[3][4]);
main()
{
   int h, k;
   int ar[3][4] = {
         {5, 10, 15, 20},
         {10, 20, 30, 40},
         {20, 40, 60, 80}
   };
   display(ar);

   for (h = 0; h < 3; h++) { add1(ar[h], 4); }     /* line a */

   printf("After line a: \n");
   display(ar);

   return 0;
}

void add1(int ar[], int size)
{
   int k;
   for (k = 0; k < size; k++)
         ar[k]++;
```

```
}
void display(int ar[3][4])
{
   int l, m;
   for (l = 0; l < 3; l++) {
         for (m = 0; m < 4; m++)
               printf("%5d", ar[l][m]);
         printf("\n");
   }
}
```

What if the for statement at 'line a' is replaced by this statement:

```
   add1(ar[0], 3 * 4);
```

3. What does the following program print?

```
#include <stdio.h>
#include <string.h>
#define M1 "How are ya, sweetie?"

char M2[40] = "Beat the clock.";
char *M3 = "chat";

int main()
{
   char words[80];
   printf(M1);
   puts(M1);
   puts(M2);
   puts(M2+1);
   gets(words);            /* user inputs : win a toy. */
   puts(words);
   scanf("%s", words+6);   /* user inputs : snoopy. */
   puts(words);
   words[3] = '\0';
   puts(words);
   while (*M3) puts(M3++);
   puts(--M3);
   puts(--M3);
   M3 = M1;
   puts(M3);
   return 0;
}
```

4. Implement the function stringncpy() that copies not more than *n* characters (characters that follow a null character are not copied) from the array pointed to by s2 to the array pointed to by s1. If the array pointed to by s2 is a string shorter than *n* characters, NULL characters are appended to the copy in the array pointed to by s1, until *n* characters in all have been written. The function prototype is given below:

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