MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG213 Data Structures – Worksheet 2

Revision
(Pointers, Pointers and Arrays, and Structures)

Purpose:
This worksheet aims to revise the following topics: pointers, the relationship between pointers and arrays, and structures.
At the end of this worksheet, you will revise:
1. how to use pointers, and how to do pointer based C programming;
2. the relation between pointers and arrays;
3. how to do dynamic memory allocation;
4. how to use structures.

1) You are developing a database of measured meteorological data for use in weather and climate research. Define a structure type measured_data_t with components site_id_num (a four-digit integer), wind_speed , day_of_month , and temperature . Each site measures its data daily, at noon local time. Write a program that inputs a file of measured_data_t records and determines the site with the greatest variation in temperature (defined here as the biggest difference between extrema) and the site with the highest average wind speed for all the days in the file. You may assume that there will be at most ten sites. Test the program on the following July daily data collected over one week at three sites:

ID	Day	Wind Speed (knots)	Temperature (deg C)
2001	10	11	30
2001	11	5	22
2001	12	18	25
2001	13	16	26
2001	14	14	26
2001	15	2	25
2001	16	14	22
3345	10	8	29
3345	11	5	23
3345	12	12	23
3345	13	14	24
3345	14	10	24

2) Microbiologists estimating the number of bacteria in a sample that contain bacteria that do not grow well on solid media may use a statistical technique called the most probable number (MPN) method. Each of five tubes of nutrient medium receives 10 ml of the sample. A second set of five tubes receives 1 ml of sample per tube, and in each of a third set of five tubes, only 0.1 ml of sample is placed. Each tube in which bacterial growth is observed is recorded as a positive, and the numbers for the three groups are combined to create a triplet such as 5-2-1, which means that all five tubes receiving 10 ml of sample show bacterial growth, only two tubes in the 1-ml group show growth, and only one of the 0.1-ml group is positive. A microbiologist would use this combination-of-positives triplet as an index to a table like the table below to determine that the most probable number of bacteria per 100 ml of the sample is 70, and 95 percent of the samples yielding this triplet contain between 30 and 210 bacteria per 100 ml.

Define a structure type to represent one row of the MPN table. The structure will include one string component for the combination-of-positives triplet and three integer components in which to store the associated most probable number and the lower and upper bounds of the 95 percent confidence range. Write a program to implement the following algorithm for gen-erating explanations of combination-of-positives triplets.

Combination of Positives	MPN Index/100 ml	95 percent Confidence Limits	
		Lower	Upper
4-2-0	22	9	56
4-2-1	26	12	65
4-3-0	27	12	67
4-3-1	33	15	77
4-4-0	34	16	80
5-0-0	23	9	86
5-0-1	30	10	110
5-0-2	40	20	140
5-1-0	30	10	120
5-1-1	50	20	150
5-1-2	60	30	180
5-2-0	50	20	170
5-2-1	70	30	210
5-2-2	90	40	250
5-3-0	80	30	250
5-3-1	110	40	300
5-3-2	140	60	360

- Load the MPN table from a file into an array of structures called mpn_table .
- Repeatedly get from the user a combination-of-positives triplet, search for it in the combination-of-positives components of mpn_table , and then generate a message such as: For 5-2-1, MPN = 70; 95% of samples contain between 30 and 210 bacteria/ml.
- Define and call the following functions.
    - load_Mpn_Table —Takes as parameters the name of the input file, the mpn_table array and its maximum size. Function opens the file, fills the mpn_table array, and closes the file. Then it returns the actual array size as the function result. If the file contains too much data, the function should store as much data as will fit, display an error message indicating that some data has been ignored, and return the array's maximum size as its actual size.
    - search —Takes as parameters the mpn_table array, its actual size, and a target string representing a combination-of-positives triplet. Returns the subscript of the structure whose combination-of-positives component matches the target or -1 if not found.

3) Write a program that aims to represent an RGB images and does some computations based on these. For this program you need to provide the following:

   a. Write a structure called **`RGB_Image`** that represents an RGB image – composed of three arrays (R, G and B) each with size of 5 by 5 pixels.

   b. Write a function called **`formRGBImage`** that takes the number of images that will be created and then it dynamically creates an array of RGB images, randomly assigns integer values between 0 and 255 for each image (R, G and B) and then returns the array back to the main function.

   c. Write a function called **`AverageOfChannels`** that takes an array structure, calculates and displays the average of R, G and B channels of each image.

Sample run can be as follows for the created 2 RGB images;

Image 1: R

26	118	60	82	118
109	136	123	77	75
11	225	230	160	182
166	168	179	119	51
35	58	102	154	97

Image 1: G

132	191	147	117	59
171	203	79	14	5
227	105	88	9	184
209	22	213	47	185
195	53	176	93	117

Image 1: B

4	185	90	24	62
64	31	78	125	24
180	216	32	192	116
59	160	14	160	150
251	203	236	224	96

Image 2: R

88	49	45	110	215
197	94	37	166	61
217	60	149	15	223
146	78	115	229	183
150	23	109	188	197

Image 2: G

179	230	53	56	223
180	91	170	218	122
62	215	40	136	136
85	182	226	76	163
41	54	115	191	106

Image 2: B

140	79	101	28	211
6	130	197	101	233
159	164	237	242	109
103	210	10	97	6
84	22	112	245	237

```
Enter the number of images you want: 2

Average of each channel of images are as follows:

Image 1: R=114, G=122, B=119
Image 2: R=126, G=134, B=131
```