Practice 9

Problem 1: Matrix Print

- Write a function that prints a 4 by 4 matrix.
- Function prototype

  void printMatrix(int inMatrix[][4]);
- Test the function in the main function.
  - In the main function, declare a 4 by 4 matrix using 2D array.
  - Fill up the matrices with random numbers (0~100).
  - Call the print function (put the random matrix as a parameter)
- Example output

```
28   93   82   83
74   61   4    21
22   9    19   73
33   55   62   98
```

Problem 2: Matrix Addition and Multiplication

- Write a program that calculates the sum and multiplication of two 4 by 4 matrices.
- Declare two 4 by 4 matrices using 2D array.
- Fill up the matrices with random numbers (0~100).
- Calculate the sum and of the matrices and save the results into new 4 × 4 matrices.
- Print out the results using the printMatrix function implemented in Problem 1.
- Example output

```
Sum:
100  194  24   98
23   13   91   122
18   110  187  162
101  45   83   92
Multiplication:
928  837  162  930
991  1827 903  192
987  9899 1872 938
991  82   934  827
```

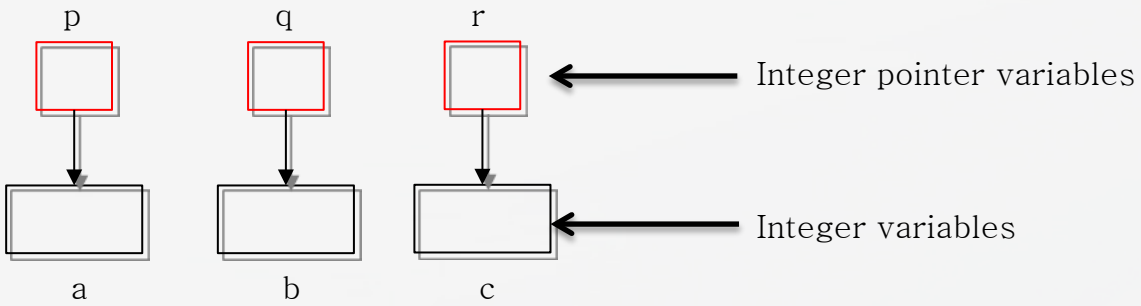
Problem 3: Diagonal Matrix

- Write a program that fills the right-to-left diagonal of a square matric with 0s., the lower right triangle with -1s, and the upper left triangle with +1s. The output of the program, assuming a 6 by 6 matrix, is shown below

1	1	1	1	1	0
1	1	1	1	0	−1
1	1	1	0	−1	−1
1	1	0	−1	−1	−1
1	0	−1	−1	−1	−1
0	−1	−1	−1	−1	−1

Problem 4: Pointer Basics

- Write a program that creates the structure shown below. The program then prompts a user to enter the data for a, b, and c. When doing this, use only p, q, and r in cin statement.



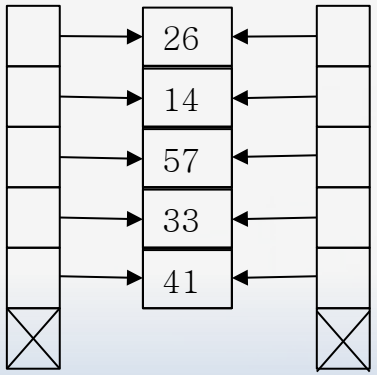
- After the data has been entered, the program reassigns the pointers so that p points to c, q points to a, and r points to b. After making the reassignments, it prints the variables using the pointers. For each variables, print both its contents and its address.

Problem 5: Function and Pointers

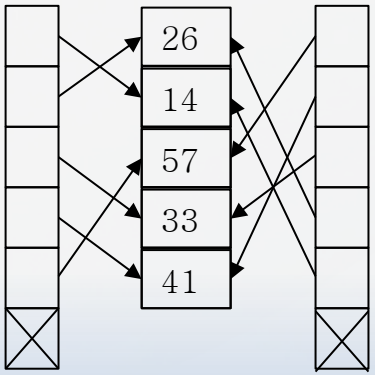
- See Program 4-8 on page 144.
- The program uses "pass by reference" technique.
- Modify the functions in the program so that it does the same task but now uses "pass by address" technique for parameter passing.

Problem 6: Array of Pointers

- Write a program that will read 10 integers from a user and place them in an array. The program then will sort the array into ascending and descending order and print the sorted list.
- The program must not change the original array or create any other integer array.
- Hint: Use two pointer arrays, as shown below. The first pointer array is rearranged to that it points to the data in ascending sequence. The second pointer array is rearranged to that is points to the data in descending sequence.



Before sorting



Ascending Descending

After sorting