Problem 1 Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.

```c
int main() {
    string cunyID = "23456789", borough = "Queens";
    int x;
    double z[2][4] = {{1.1, 1.11, 2.5, 5.7}, {1,2,3,4}};

    // a. The function firstDigit extracts the first digit.
    x = firstDigit(cunyID); // (a)
    cout << x << endl; // prints 2

    // b. The function randomElement selects a random element in a 2d-array.
    cout << randomElement(z, 2, 4) << endl; // (b)
    // could print 1.1 or 3 or other elements

    // c. The function cutVowels removes any vowels from a string.
    cutVowels( borough ); // (c)
    cout << borough << endl; // prints Qns

    // d. A mystery function.
    mystery( mystery(z[0][0], z[0][1]), mystery(z[0][0], z[0][2]) ); // (d)

    // e. An enigma
    enigma( enigma(mystery(z[0][0], z[0][1])) ); // (e)

    return 0;
}
```

(a) Title line for **firstDigit** as called at the line marked (a).
**Answer:** int firstDigit(string s)

(b) Title line for **randomElement** as called at the line marked (b).
**Answer:** double randomElement(double z[][4], int r, int c)

(c) Title line for **cutVowels** as called at the line marked (c).
**Answer:** void cutVowels(string &s)

(d) Title line for **mystery** as called at the line marked (d).
**Answer:** double mystery(double a, double b)

(e) Title line for **enigma** as called at the line marked (e).
**Answer:** double enigma(double u)
Problem 2  Write blocks (that is the code) for the functions that are called by the following main program. No block requires more than 5 lines of code. Excessively long and complicated blocks will not receive credit. Do not supply the title lines for the functions. These are given to you.

```cpp
int main() {
    srand(time(0));
    int a = 2, b = 1, c = 3;
    int array[5] = { 3, 1, 4, 1, 5};
    cout << average(a, b, c) << endl;  // (a) prints 2.0
    cout << smallest(a, b, c) << endl; // (b) prints 1
    cycle(a, b, c);                   // (c) a,b,c get old values of b,c,a
    cout << a << " " << b << " " << c << endl;  // prints 1 3 2
    cout << randomElement(array, 5) << endl; // (d) prints a randomly selected element of the array
    multiPrint("Hello", 5, 3);            // (e) print Hello 5 times with 3 copies per line, here as:
                                         // Hello Hello Hello
                                         // Hello Hello
                                         // Hello Hello
    return 0;
}
```

(a) Write the block for the function called at the line marked (a). It has title line:

double average(int x, int y, int z)
it returns the average of its 3 parameters.

**Answer:**

```cpp
{
    return (x + y + z) / 3.0;
}
```

(b) Write the block for the function called at the line marked (b). It has title line:

int smallest(int x, int y, int z)
it returns the smallest of 3 parameters.

**Answer:**

```cpp
{
    int ans = x;
    if (y < ans) ans = y;
    if (z < ans) ans = z;
    return ans;
}
```

(c) Write the block for the function called at the line marked (c). It has title line:

void cycle(int &x, int &y, int &z)
it changes parameters x, y, z to become what y, z and x were originally.

**Answer:**

```cpp
{
    int temp = x;
    x = y;
    y = z;
    z = temp;
}
```

(d) Write the block for the function called at the line marked (d). It has title line:

int randomElement(int array[], int cap)
it uses a standard random number function to return a random element from an array.

**Answer:**
(e) Write the block for the function called at the line marked (e). It has title line:

```cpp
void multiPrint(string s, int n, int m)
```

it prints n copies of s in batches of m to a line.

**Answer:**

```cpp
{
    for (int i = 1; i <= n; i++) {
        cout << s << " ";
        if (i % m == 0) cout << endl;
    }
    cout << endl;
}
```
Problem 3 Consider the following C++ program. Which is compiled and run by a user with the command
./a.out 88845678 Freddy
Here instead of ID8 the user types your 8-digit CUNY ID number and instead of NAME the user types your first name. So for example if your NAME and ID are Freddy and 88845678 the user types
./a.out 88845678 Freddy
The code for the functions firstCharacter and thirdDigit has been omitted, but they return the 1st character from a string or the 3rd digit from a string of digits. For example firstCharacter("12345678") returns the character '1' and thirdDigit("12345678") returns the integer 3.

```cpp
int main(int argc, char *argv[]) {
    string name = argv[2], cunyID = argv[1];
    int x = thirdDigit(cunyID);
    char c = firstCharacter(name);
    char d = firstCharacter(cunyID);

    cout << c << d << x << endl; // line (a)
    for (int y = x; y > 1; y--) cout << y; cout << endl; // line (b)
    cout << (char) (cunyID[2] - x) << endl; // line (c)
    cout << argv[0] << endl; // line (d)
    cout << argc << endl; // line (e)
    return 0;
}
```

The following are answers for student: 88845678 Freddy. Your answers will be different.
(a) What is the output from the instruction beginning on line (a)?
**Answer:**
F88

(b) What is the output from the instruction beginning on line (b)?
**Answer:**
8765432

(c) What is the output from the instruction beginning on line (c)?
**Answer:**
0

(d) What is the output from the instruction beginning on line (d)?
**Answer:**
./a.out

(e) What is the output from the instruction beginning on line (e)?
**Answer:**
3
Problem 4 Let \( N \) be the number formed by the first 2 digits of your CUNY ID number. For example if your ID number is 25677666 then \( N \) is 25. Write a function called `swapColumns` that swaps two columns of a 2-dimensional array of integers with \( N \) columns. The function should use 5 parameters as follows: the array name, the number of rows, the number of columns and the numbers of the two columns to be swapped. Your function should check that the column numbers are legal. If they are not it should print the message: **Illegal columns**. Excessively long solutions that use more than 12 lines of code may lose points. A program that uses the function `swapColumns` follows. Your code must use an actual number in place of \( N \).

// This example assumes \( N \) is 5. You will use a different value of \( N \).
int main() {
    int x[4][5] = {{0,1,2,3,4}, {3,4,5,6,7}, {6,7,8,9,10}, {9,10,11,12,13}};
    swapColumns(x, 4, 5, 0, 1); // swaps the first two columns of x
    cout << x[0][0] << endl; // prints 1
    swapColumns(x, 4, 5, 0, 9); // prints Illegal columns
    return 0;
}

Answer:

void swapColumns(int a[][5], int rows, int cols, int i, int j) {
    if ( i < 0 || j < 0 || i >= cols || j >= cols)
        cout << "Illegal columns" << endl;
    else for (int r = 0; r < rows; r++) {
        int temp = a[r][i];
        a[r][i] = a[r][j];
        a[r][j] = temp;
    }
}
Problem 5  

The recursive function \( \text{changeDigits}(x, d, c) \) changes a positive integer parameter \( x \) so that every digit that matches \( d \) is changed to match \( c \). For example, if \( \text{cunyID} \) is a variable that stores your CUNY ID number then \( \text{changeDigits}(\text{cunyID}, 5, 6) \) changes every 5 in the your ID number to a 6.

An implementation of this function with parts of the code covered up is given below. There is also a main program that uses it.

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```
PART (a) {
    if (x == 0) PART (b);
    int lastDigit = x % 10;
    if ( lastDigit == d ) PART (c);
    int y = x / 10;
    PART (d);
    x = PART (e);
    return;
}

int main() {
    int cunyID = 88877555, y = 911;
    changeDigits(cunyID, 5, 6);
    cout << cunyID << endl;  // prints 88877666
    changeDigits(y, 3, 2);
    cout << y << endl;  // prints 911
    changeDigits(y, 1, 2);
    cout << y << endl;  // prints 922
    return 0;
}
```

(a) Give a replacement for PART (a) as the title line:

**Answer:** PART (a) is \( \text{void changeDigits(int} \& x, \text{ int} \ d, \text{ int} \ c) \)

(b) Give a replacement for PART (b) as the base case of recursion:

**Answer:** PART (b) is \( \text{return} \)

(c) Give a replacement for PART (c) to use the correct last digit of the answer we want:

**Answer:** PART (c) is \( \text{lastDigit} = c \)

(d) Give a replacement for PART (d) to change digits in \( y \):

**Answer:** PART (d) is \( \text{changeDigits(y, d, c);} \)

(e) Give a replacement for PART (e) to change the digits in \( x \):

**Answer:** PART (e) is \( 10 \ast y + \text{lastDigit} \)
Problem 6  
The following program draws a picture of a triangle in an output file. The name of the output file must be set to match your first name. For example, if your name is Freddy then the output file is called Freddy. The number of rows shown in the triangle is \(N\) where \(N\) is the number formed by the first 2 digits of your CUNY 8-digit ID number. For example, if your number is "44456789" your triangle would have 44 rows (because the first 2 digits are 44). The width of the first row is \(2N - 1\) and the width of the last row is 1. The character used to print the triangle is the last digit \(D\) of your CUNY 8-digit ID number.

For example, if \(N\) is 4, and \(D\) is 9 the output would appear as follows:

```
99999999
999999999
9999999999999999
```

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```c++
int main() {
    PART (a) file;
    PART (b)
    for (int row = 0; PART (c); row++) {
        for (int col = 1; PART (d); col++)
            if (col <= PART (e))
                file << PART (f);
            else file << PART (g);
        file << PART (h);
    }
    PART (i);
    PART (j);
}
```

These answers use \(N\) and \(D\) in place of the numbers that you should use.

(a) Give a replacement for PART (a) to declare a file variable:

**Answer:** PART (a) is `ofstream`

(b) Give a replacement for PART (b) to open the file:

**Answer:** PART (b) is `file.open("Freddy");`

(c) Give a replacement for PART (c) to count rows:

**Answer:** PART (c) is `row < N`

(d) Give a replacement for PART (d) to count columns:

**Answer:** PART (d) is `col <= 2 * N - 1`

(e) Give a replacement for PART (e) to decide what to print in a column:

**Answer:** PART (e) is `2 * row`

(f) Give a replacement for PART (f) to print:

**Answer:** PART (f) is ` " "`

(g) Give a replacement for PART (g) to print:

**Answer:** PART (g) is `D`

(h) Give a replacement for PART (h) to print:

**Answer:** PART (h) is `endl`

(i) Give a replacement for PART (i) to close the file:

**Answer:** PART (i) is `file.close();`

(j) Give a replacement for PART (j) to end the program:

**Answer:** PART (j) is `return 0`