The following problems have been used on previous final exams in CS111. The exam will have 6 problems.

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.

```cpp
int main() {
    int y = 13579, z[5] = {1, 3, 5, 7, 248};
    // the function smallDigit finds the smallest digit in a parameter, here it is 2
    cout << smallDigit(z[4]) << endl; // (a)
    // the function sumTheDigits replaces a parameter by the sum of its digits
    sumTheDigits(y); // (b)
    cout << y << endl; // prints 25
    // the function sumBoth adds all the elements in two arrays
    cout << sumBoth(z, z, 5) << endl; // (c)
    cout << mystery(z[0], z) << endl; // (d)
    puzzle(mystery(z[0], z), puzzle('a', false)); // (e)
    return 0;
}
```

(a) Title line for smallDigit as called at the line marked (a).
Answer: `int smallDigit(int x)`

(b) Title line for sumTheDigits as called at the line marked (b).
Answer: `void sumTheDigits(int &x)`

(c) Title line for sumBoth as called at the line marked (c).
Answer: `int sumBoth(int x[], int y[], int c)`

(d) Title line for mystery as called at the line marked (d).
Answer: `char mystery(int x, int y[])`

(e) Title line for puzzle as called at the line marked (e).
Answer: `bool puzzle(char x, bool y)`

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 << randChoice(a, b) << endl; // (a) prints either 2 or 1
    cout << largest(a, b, c) << endl; // (b) prints 3
    duplicate(a, b, c); // (c) b and c become copies of a (ie 2)
    cout << a << " " << b << " " << c << endl; // prints 2 2 2
    cout << average(array, 5) << endl; // (d) prints the average: 2.8
    multiPrint("Hello", 4, 3); // (e) print Hello 4 times with 3 copies per line, here as:
                              // Hello Hello Hello
                              // Hello
    return 0;
}
```

(a) Write the block for the function called at the line marked (a). It has title line:
double randChoice(int x, int y)
it uses a standard random number function to return a random choice of its 2 parameters.

Answer:
{
    if (rand() % 2 == 0) return x;
    return y;
}

(b) Write the block for the function called at the line marked (b). It has title line:
int largest(int x, int y, int z)
it returns the largest of 3 parameters.

Answer:
{
    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 duplicate(int x, int &y, int &z)
it changes parameters y, z to become what x was originally.

Answer:
{
    y = x;
    z = x;
    return;
}

(d) Write the block for the function called at the line marked (d). It has title line:
double average(int array[], int cap)
returns the average of the array entries.

Answer:
{
    int sum = 0;
    for (int i = 0; i < cap; i++) sum += array[i];
    return (double) sum / cap;
}

(e) Write the block for the function called at the line marked (e). It has title line:
void multiPrint(string s, int n, int m)
it prints n copies of s in batches of m to a line.

Answer:
{
    for (int i = 1; i <= n; i++) {
        cout << s << " ";
        if (i % m == 0) cout << endl;
    }
    cout << endl;
    return;
}
Problem 3  Consider the following C++ program.

When the following program is run the user types your 8-digit CUNY ID number and types your first name for the input. The output that you predict should be based on these inputs. It would be a bad mistake to use another student’s name or ID!

The code for the functions lastDigit and copies has been omitted, but they return the last digit of an integer and a string formed from a specified number of copies of a string argument. For example lastDigit(88877776) returns 6 and copies("Freddy", 2) returns "FreddyFreddy". (And copies("Freddy", 0) returns the empty string ".")

```c++
int main() {
    string name; int cunyID;
    cout << "Enter your name and CUNY ID: ";
    cin >> name >> cunyID;  // the user types your first name and CUNY ID
    cout << name << endl ; // line (a)
    int x = lastDigit(cunyID /100);
    string c = copies(name, x % 3 + 1);

    cout << c << endl ;  // line (b)
    cout << copies("Freddy", x / x % x) << endl ; // line (c)
    cout << copies("Freddy", 2).erase(4,2) << endl ; // line (d)
    for (int y = x; y > 0; y--) cout << y; cout << endl ;  // line (e)

    return 0;
}
```

Enter your name and CUNY ID: (a) What is the output from the instruction beginning on line (a)?

**Answer:**

Alex
12345678

(Students will have different answers here.)

(b) What is the output from the instruction beginning on line (b)?

**Answer:**

Alex

(The name from part (a) is repeated N times where N is found by taking the remainder when the 6th digit of the ID number is divided by 3 and adding 1. The number of repeats will always be 1, 2 or 3.)

(c) What is the output from the instruction beginning on line (c)?

**Answer:**

Freddy

(d) What is the output from the instruction beginning on line (d)?

**Answer:**

FredFreddy

(e) What is the output from the instruction beginning on line (e)?

**Answer:**
(The answer here will depend on the student’s CUNY ID number. It begins with the 6th digit of the CUNY ID number (if it is at least 1) and follow it with decreasing digits ending at 1. If the 6th digit is 0, the answer here is blank.)

Problem 4 The function called cutDuplicates has two parameters array and cap. The first is an array and the second represents the capacity of the array. The function decreases every array entry that was initially equal to its left hand neighbor. (The left hand neighbor of the entry with index i + 1 has index i.)

For example, if cutDuplicates is applied to an array with entries \{2, 2, 2, 1, 1, 1, 3, 3\} it is changed to have entries \{2, 1, 1, 0, 0, 3, 2\}.

(a) What is your 8-digit CUNY ID number? Answer: 12345678
(This will be different for other students.)

(b) If an array with capacity 8 stores the digits of your CUNY ID number (in order) what would the array be changed to after it is used as input for the function cutDuplicates? Answer: \{1,2,3,4,5,6,7,8\}
(This will be found by applying the function to the digits in the answer to (a).)

(c) Write a C++ implementation of the function cutDuplicates. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function cutDuplicates follows.

```cpp
int main() {
    int x[8] = {2, 2, 2, 1, 1, 1, 3, 3};
    cutDuplicates(x, 8);
    for (int i = 0; i < 8; i++)
        cout << x[i];  // output 2110032
    cout << endl;
    return 0;
}

Answer:

void cutDuplicates(int array[], int cap) {
    for (int i = cap - 1; i > 0; i--)
        if (array[i] == array[i - 1]) array[i] --;
    return;
}
```

Problem 5 The recursive function addZeros(x) returns a number formed from x by inserting a 0 between every pair of adjacent non-zero digits. For example, addZeros(1234) returns 1020304, addZeros(1023) returns 10203, addZeros(10023) returns 100203 and addZeros(10203) returns 10203. 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 (PART (b)) return x;
    int y = PART (c);
    if ( (x % 10 == 0) PART (d) y % 10 == 0)
        return PART (e);
    return PART (f);
```

Problem 5 The recursive function addZeros(x) returns a number formed from x by inserting a 0 between every pair of adjacent non-zero digits. For example, addZeros(1234) returns 1020304, addZeros(1023) returns 10203, addZeros(10023) returns 100203 and addZeros(10203) returns 10203. 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 (PART (b)) return x;
    int y = PART (c);
    if ( (x % 10 == 0) PART (d) y % 10 == 0)
        return PART (e);
    return PART (f);
```
```cpp
int main() {
    int cunyID = 88877005;
    cout << addZeros(cunyID / 10000) << endl; // prints 8080807
    cout << addZeros(cunyID % 10000) << endl; // prints 7005
    return 0;
}
```

(a) Give a replacement for PART (a) as the title line, the parameter name should be x:

**Answer:** PART (a) is `int addZeros(int x)`

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

**Answer:** PART (b) is `x < 10`

(c) Give a replacement for PART (c) as the recursive call:

**Answer:** PART (c) is `addZeros(x / 10)`

(d) Give a replacement for PART (d) to detect when no more 0s must be added:

**Answer:** PART (d) is `||`

(e) Give a replacement for PART (e) to return an answer in this case:

**Answer:** PART (e) is `10 * y + x % 10`

(f) Give a replacement for PART (f) to return an answer in the remaining case:

**Answer:** PART (f) is `100 * y + x % 10`

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 last name. For example, if your name is Harry Potter then the output file is called Potter. The number of rows shown in the triangle is N where N is the number formed by the first 2 digits of your 8-digit CUNY ID number. For example, if your number is "04456789" your triangle would have 4 rows (because the first 2 digits are 04). The first row has width 1, the second has width 3 and row widths continue to increase in steps of 2. The rows are arranged so that their centers are aligned. 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:

```
  9
  999
  999999
  99999999
```

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.

// This is Harry Potter’s program but uses N and D to stand for numbers from his ID.
int main() {
    PART (a) file;
    PART (b)
    for (int row = 1; PART (c); row++) {
        for (int col = 1; PART (d); col++)
    PART (e)
        for (int col = 1; PART (f); col++)
    PART (g)
        PART (h);
    }
    PART (i);
    PART (j);
}

(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("Potter");
(This will use the student’s last name instead.)
(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 initial blank columns:
Answer: PART (d) is  col <= N - row
(e) Give a replacement for PART (e) to print each blank column
Answer: PART (e) is  file << " ";
(f) Give a replacement for PART (f) to count digits to print:
Answer: PART (f) is  col <= 2*row - 1
(g) Give a replacement for PART (g) to print each digit
Answer: PART (g) is  file << D;
(h) Give a replacement for PART (h) to finish a row:
Answer: PART (h) is  file << 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

Problem 7  Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.

```cpp
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)
Problem 8 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
    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:

```c
int randomElement(int array[], int cap)
```

It uses a standard random number function to return a random element from an array.

Answer:

```c
{
    return array[rand() % cap];
}
```

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

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

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

Answer:

```c
{
    for (int i = 1; i <= n; i++) {
        cout << s << " ";
        if (i % m == 0) cout << endl;
    }
    cout << endl;
}
```

**Problem 9** 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.**

```c
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; // (d) 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
    return 0;
}
```

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

```c
double average(int x, int y, int z)
```

It returns the average of its 3 parameters.

Answer:

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

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

```c
int smallest(int x, int y, int z)
```

It returns the smallest of 3 parameters.

Answer:
{  
  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:
{
  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:
{
  return array[rand() % cap];  
}

(e) Write the block for the function called at the line marked (e). It has title line:
void multiPrint(string s, int n, int m)
it prints n copies of s in batches of m to a line.
Answer:
{
  for (int i = 1; i <= n; i++) {
    cout << s << " ";  
    if (i % m == 0) cout << endl;  
  }  
  cout << endl;  
}

Problem 10 Consider the following C++ program. Which is compiled and run by a user with the command
./a.out ID8 NAME
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.

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 11  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;
}
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 12
The recursive function 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 cunyID is a variable that stores your CUNY ID number then changeDigits(cunyID, 5, 6) changes every 5 in 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 void changeDigits(int &x, int d, int c)

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

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

(d) Give a replacement for PART (d) to change digits in y:
Answer: PART (d) is changeDigits(y, d, c);

(e) Give a replacement for PART (e) to change the digits in x:
Answer: PART (e) is 10 * y + lastDigit
Problem 13  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
999999
99999
9999
999
99
9
```

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`

Problem 14  Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.
Problem 15 Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.

```cpp
int main() {
    int a[3] = {10, 20, 30};
    double x;
    string b = "thing1", c = "thing2";

    // set x to be the average of a[0], a[1] and a[2]
    x = average(a[0], a[1], a[2]); // (a)
    cout << x << endl;
    // print the average to screen
    average(a, 3); // (b)
    // print to screen: The strings are thing1 and thing2
    cout << "The strings are " << whatStrings(b, c) << endl; // (c)
    if (mystery(mystery(function()))) // (d)
        cout << mystery(function()) << endl; // (e)
    return 0;
}
```

(a) Title line for `average` as called at the line marked (a).
Answer: `double average(int a, int b, int c)`

(b) Title line for `average` as called at the line marked (b).
Answer: `void average(int a[], int c)`

(c) Title line for `whatStrings` as called at the line marked (c).
Answer: `string whatStrings(string a, string b)`

(d) Title line for `function` as called at the line marked (d).
Answer: `bool function()`

(e) Title line for `mystery` as called at the line marked (e).
Answer: `bool mystery(bool b)`

Problem 16 Consider the following C++ program.

```cpp
int main() {
    cout << words[3].substr(2) << endl; // line (a)
    words[3].replace(1,2,"8");
    cout << words[3] << endl; // line (b)
    cout << words[2][2] << endl; // line (c)
    cout << words[1].rfind("1") << endl; // line (d)
    for (int i = 1; i <= 2; i++) cout << words[i].substr(1,i); // line (e)
    cout << endl;
    return 0;
}
```

(a) What is the output at line (a)?
Answer:
19

(b) What is the output at line (b)?
Answer:
(c) What is the output at line (c)?
Answer:

y

(d) What is the output at line (d)?
Answer:

2

(e) What is the output at line (e)?
Answer:

1ay

Problem 17  Write blocks of code to perform the functions used in the following main program. Your blocks must match the given title lines. Each block should be a short function that is written only in the designated answer space.

```c
int main() {
    double x = 1.0, y = -0.5;
    int a[5] = {81, 3, 4, 1024, 0};
    // (a) Return true if both parameters are positive. Here NOT is printed.
    if (!bothPositive(x, y)) cout << "NOT" << endl;
    // (b) Swap the parameters if the 2nd is larger. Here 43 is printed.
    makeDecrease(a[1], a[2]); cout << a[1] << a[2] << endl;
    // (c) Return true if array contains a 0 entry. Here YES is printed.
    if (hasZero(a, 5)) cout << "YES\n";
    // (d) Return minimum number of coins to make given amount in change. Here 5.
    cout << numberCoins(a[0]) << endl;
    // (e) Returns the number of digits in a positive parameter. Here 4 is printed.
    cout << numberDigits(a[3]) << endl;
    return 0;
}

Answer:

(a)

bool bothPositive(double x, double y) {
    return x > 0 && y > 0;
}

(b)

void makeDecrease(int &x, int &y) {
    if (x < y) {
        double temp = x;
        x = y;
        y = temp;
    }
}
```
bool hasZero(int x[], int capacity) {
    for (int i = 0; i < capacity; i++)
        if (x[i] == 0) return true;
    return false;
}

int numberCoins(int amount) {
    int ans = amount / 25;
    amount = amount % 25;
    ans += amount / 10;
    amount = amount % 10;
    return ans + amount / 5 + amount % 5;
}

int numberDigits(int x) {
    if (x < 10) return 1;
    return 1 + numberDigits(x / 10);
}

Problem 18  Write a function called maximumGap that finds the maximum difference between corresponding entries in two arrays (that have the same capacity). There are 3 parameters, the 2 arrays and their capacity. If the arrays store 9, 0, 3 and 4, 7, 6 the maximum gap is 7 (between the second elements of the two arrays).
Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function maximumGap follows.

int main() {
    int pi[8] = {3, 1, 4, 1, 5, 9, 2, 6};
    int e[8] = {2, 7, 1, 8, 2, 8, 1, 8};
    cout << maximumGap(pi, e, 8) << endl; // prints 7 the gap between 1 and 8
    return 0;
}
Problem 19   Write a function called *maskDigits* that takes a positive integer parameter and returns a string of * symbols with one * for each digit. For example, if the parameter is 19683 the function will return. *****. Your function can return any convenient value if it is given illegal input.

Excessively long solutions that use more than 6 lines of code may lose points. A program that uses the function *maskDigits* follows.

```
int main() {
    cout << maskDigits(19683) << endl; // prints *****
    cout << maskDigits(19) << endl; // prints **
    cout << maskDigits(1) << endl; // prints *
    return 0;
}
```

Answer:

```
string maskDigits(int a) {
    if (a <= 0) return "";
    return "*" + maskDigits(a / 10);
}
```

Problem 20   Write a program that reads 100 single digit numbers from the user. It then prints them as in a square with 10 columns so that the first 10 numbers occupy the first column, the next 10 occupy the next column and so on.

Excessively long solutions that use more than 15 lines of code may lose points.

Answer:

```
int main() {
    int x[10][10];
    cout << "Enter 100 digits: ";
    for (int c = 0; c < 10; c++)
        for (int r = 0; r < 10; r++) cin >> x[r][c];
    for (int r = 0; r < 10; r++) {
        for (int c = 0; c < 10; c++) cout << x[r][c] << " ";
        cout << endl;
    }
    return 0;
}
```

Problem 21   Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.

```
int main() {
    int nn = 5, num[3] = {0, 1, 2};
    double rl[3] = {1.9, 2.3, 3.0};
    char ch = 'b';

    nn = add(ch, 1); // (a)
    printProduct(num[2], rl[1]); // (b)
    if (qn(rl, 3)) cout << "Yes\n"; // (c)
    d(nn, 2); // (d)
    e(d(nn, rl[1]), e('n', e('b', rl[2]))); // (e)
    return 0;
}
```
Problem 22  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() {
    int nn = 5, num[3] = {0, 1, 2};
    double rl[3] = {1.9, 2.3, 3.0};
    char ch = 'b';

    swap(num[1], nn); // (a)
    cout << num[1] << nn << endl; // prints 51
    remove(nn, num, 3); // (b)
    // the next line prints 3.1415926
    cout << qn(rl, 3) << "\n"; // (c)
    d(num, 2); // (d)
    e(d(num, 1), e(true, e(true, ch))); // (e)
    return 0;
}
```

(a) Title line for swap as called at the line marked (a).
Answer: void swap(int &x, int &y)
(b) Title line for remove as called at the line marked (b).
Answer: void remove(int x, int y[], int z)
(c) Title line for qn as called at the line marked (c).
Answer: double qn(double x[], int c)
(d) Title line for d as called at the line marked (d).
Answer: char d(int x[], double y)
(e) Title line for e as called at the line marked (e).
Answer: double e(char x, double y)

Problem 23  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() {
    double xx = 5, xVals[3] = {0, 1, 2};
    int ins[3] = {1, 2, 3};
    string stg = "b";

    xx = combine(stg, 1); // (a)
    combine(xVals[2], ins[1]); // (b)
    if (partC(ins, 3)) cout << "Yes\n"; // (c)
    d(xx, 2); // (d)
    e(d(xx, ins[1]), e((char) e('b', ins[2]), 0)); // (e)
    return 0;
}
```

(a) Title line for swap as called at the line marked (a).
Answer: void swap(int &x, int &y)
(b) Title line for remove as called at the line marked (b).
Answer: void remove(int x, int y[], int z)
(c) Title line for qn as called at the line marked (c).
Answer: double qn(double x[], int c)
(d) Title line for d as called at the line marked (d).
Answer: bool d(int x[], int y)
(e) Title line for e as called at the line marked (e).
Answer: char e(bool x, char y)
(a) Title line for **combine** as called at the line marked (a).
**Answer:** `double combine(string x, int y)`
(b) Title line for **combine** as called at the line marked (b).
**Answer:** `void combine(double x, int y)`
(c) Title line for **partC** as called at the line marked (c).
**Answer:** `bool partC(int x[], int c)`
(d) Title line for **d** as called at the line marked (d).
**Answer:** `char d(double x, int y)`
(e) Title line for **e** as called at the line marked (e).
**Answer:** `int e(char x, int y)`

**Problem 24** 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() {
    int nn = 5, num[3] = {0, 1, 2};
    double rl[3] = {1.9, 2.3, 3.0};
    string ch = "b";

    swap(rl[1], rl[2]); // (a)
    reOrder(num, 3); // (b)
    // the next line prints No
    cout << qn(rl, 3) << "\n"; // (c)
    d(num, 2); // (d)
    if (d(num, e(e(e(nn))))) cout << "Yes\n"; // (e)
    return 0;
}
```

(a) Title line for **swap** as called at the line marked (a).
**Answer:** `void swap(double &x, double &y)`
(b) Title line for **reOrder** as called at the line marked (b).
**Answer:** `void reOrder(int y[], int z)`
(c) Title line for **qn** as called at the line marked (c).
**Answer:** `string qn(double x[], int c)`
(d) Title line for **d** as called at the line marked (d).
**Answer:** `bool d(int x[], int y)`
(e) Title line for **e** as called at the line marked (e).
**Answer:** `int e(int x)`

**Problem 25** 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() {
    int nn = 5, num[3] = {0, 1, 2};
    double rl[3] = {1.9, 2.3, 3.0};
    char ch = 'b';

    nn = add(ch, 1); // (a)
    printProduct(num[2], rl[1]); // (b)
    if (qn(rl, 3)) cout << "Yes\n"; // (c)
    d(2); // (d)
    nn = e(d(rl[1])) + e(ch); // (e)
    return 0;
}
```
Problem 26  Consider the following C++ program.

```cpp
void a(int &x, int y) {
    x = y;
    y = x + 1;
}

int b(int &x, int y) {
    y = x + 1;
    x = y;
    return y;
}

void c(int x[], int y) {
    if (y < 1) return;
    cout << x[y - 1] - y;
    c(x, y - 1);
}

int main() {
    int x[6] = {1, 2, 3, 4, 5, 6};
    int y[3] = {5, 4, 3};
    cout << x[y[1]] % x[y[1]] << endl; // line (a)
    a(x[0], x[1]);
    cout << x[0] << x[1] << endl; // line (b)
    cout << b(x[2], x[3]) << endl; // line (c)
    c(y, 3); cout << endl; // line (e)
}
```

(a) What is the output at line (a)?
**Answer:**
3

(b) What is the output at line (b)?
**Answer:**
22

(c) What is the output at line (c)?
**Answer:**
4
(d) What is the output at line (d)?
Answer:
44

(e) What is the output at line (e)?
Answer:
024

Problem 27  Consider the following C++ program.

```cpp
void a(int x, int &y) {
    x = y;
    y = x + 1;
}
int b(int &x, int &y) {
    y = x + 1;
    x = y;
    return y - 2;
}
void c(int x[], int y) {
    if (y < 1) return;
    c(x, y - 1);
    cout << x[y - 1] * y;
}

int main() {
    int x[6] = {2, 3, 4, 5, 6, 7};
    int y[4] = {6, 5, 4, 3};
    cout << x[y[1]] % y[x[1]] << endl;  // line (a)
a(x[0], x[1]);
    cout << x[0] << x[1] << endl;     // line (b)
    cout << b(x[2], x[3]) << endl;    // line (c)
c(y, 3); cout << endl;            // line (e)
}
```

(a) What is the output at line (a)?
Answer:
1

(b) What is the output at line (b)?
Answer:
24

(c) What is the output at line (c)?
Answer:
3
Problem 28  Consider the following C++ program.

```cpp
void a(int &x, int y) {
    x = y;
    y = x + 1;
}

int b(int &x, int y) {
    y = x + 1;
    x = y;
    return y;
}

void c(int x[], int y) {
    if (y < 1) return;
    cout << x[y - 1] - y;
    c(x, y - 1);
}

int main() {
    int x[6] = {0, 2, 4, 6, 8, 10};
    int y[3] = {6, 4, 2};
    cout << y[x[1]] % x[y[1]] << endl;  // line (a)
    a(x[0], x[1]);
    cout << x[0] << x[1] << endl;       // line (b)
    cout << b(x[2], x[3]) << endl;     // line (c)
    c(y, 3); cout << endl;             // line (e)
}
```

(a) What is the output at line (a)?
Answer:
2

(b) What is the output at line (b)?
Answer:
22

(c) What is the output at line (c)?
Answer:
5

(d) What is the output at line (d)?
Answer:
Problem 29  Consider the following C++ program.

```cpp
void a(int &x, int &y) {
    x = y;
    y = x + 1;
}
int b(int &x, int y) {
    y = x + 1;
    x = y;
    return y - 2;
}
void c(int x[], int y) {
    if (y < 1) return;
    c(x, y - 1);
    cout << x[y - 1] * y;
}

int main() {
    int x[6] = {4, 3, 2, 1, 0, -1};
    int y[4] = {2, 1, 0, -1};
    cout << x[y[1]] % y[x[1]] << endl; // line (a)
a(x[0], x[1]);
    cout << x[0] << x[1] << endl; // line (b)
cout << b(x[2], x[3]) << endl; // line (c)
c(y, 4); cout << endl; // line (e)
}
```

(a) What is the output at line (a)?
**Answer:**
0

(b) What is the output at line (b)?
**Answer:**
34

(c) What is the output at line (c)?
**Answer:**
1

(d) What is the output at line (d)?
**Answer:**
31
Problem 30  Consider the following C++ program.

```cpp
void a(int &x, int y) {
    x = y;
    y = x + 1;
}
int b(int &x, int y) {
    y = x + 1;
    x = y;
    return y;
}
void c(int x[], int y) {
    if (y < 1) return;
    cout << x[y - 1] - y;
    c(x, y - 1);
}
int main() {
    int x[6] = {3, 2, 1, 6, 5, 4};
    int y[3] = {3, 4, 5};
    cout << y[x[1]] % x[y[1]] << endl; // line (a)
    a(x[0], x[1]);
    cout << x[0] << x[1] << endl; // line (b)
    cout << b(x[2], x[3]) << endl; // line (c)
    c(y, 3); cout << endl; // line (e)
}
```

(a) What is the output at line (a)?

Answer:

0

(b) What is the output at line (b)?

Answer:

22

(c) What is the output at line (c)?

Answer:

2

(d) What is the output at line (d)?

Answer:

26

(e) What is the output at line (e)?

Answer:
Problem 31  Write verb:C++: statements to carry out the following tasks. Do not write complete programs. Each answer should be at most 3 lines of C++. Assume the following variables have been declared and initialized with positive values.

    int x, y;

(a) Print 12 copies of the word Hello on a single line of output.
Answer:

    for (int c = 1; c <= 12; c++) cout << "Hello ";
    cout << endl;

(b) Print the remainder when variable x is divided by variable y.
Answer:

    cout << x % y << endl;

(c) Print the square root of 19. Use a C++ function for the calculation.
Answer:

    cout << sqrt(19.0) << endl;

(d) Print a random number in the range 23 to 34, inclusive. Use a C++ function.
Answer:

    cout << rand() % 12 + 23 << endl;

(e) Print the digits of the variable x backwards. So if x is 25, print 52.
Answer:

    while (x > 0) {
        cout << x % 10;
        x = x / 10;
    }

Problem 32  Write a program that reads a positive integer n from the user. If the user gives a non-positive value for n the program should terminate immediately. Otherwise, the program should print a triangle with n rows that looks like the following sample when n is 4.

*   *
* * *
* * * *
* * * * *

The triangle should be made with * and blank symbols with one extra * per row. The *s should be separated by blanks and the picture should be symmetric about a vertical line through its middle.
Excessively long solutions that use more than 20 lines of code may lose points.
Answer:

```cpp
    int n;
    cin >> n;
    if (n <= 0) return;

    for (int i = 1; i <= n; ++i) {
        for (int j = n - i; j > 0; --j) cout << " ";
        for (int k = 1; k <= i; ++k) cout << "*";
        cout << endl;
    }
```
int main() {
    int n;
    cout << "Enter a non-negative number n: ";
    cin >> n;
    if (n <= 0) return 0;

    for (int r = 0; r < n; r++) {
        for (int c = -(n - 1); c <= n - 1; c++)
            if (c < -r || c > r || (c + r) % 2 != 0) cout << " ";
            else cout << "*";
        cout << endl;
    }
    return 0;
}

Alternative solution:

void triangle(int n, int indent) {
    if (n == 0) return;
    triangle(n - 1, indent + 1);
    for (int c = 1; c <= indent; c++) cout << " ";
    for (int c = 1; c <= n; c++) cout << "* ";
    cout << endl;
}

int main() {
    int n;
    cout << "Enter a non-negative number n: ";
    cin >> n;
    if (n <= 0) return 0;
    triangle(n, 0);
    return 0;
}

Problem 33  Write a program that reads a positive integer n from the user. If the user gives a non-positive value for n the program should terminate immediately. Otherwise, the program should print an upside down triangle with n rows that looks like the following sample when n is 4.

X X X X
X X X
X X
X

The triangle should be made with X and blank symbols with one fewer X per row. The Xs should be separated by blanks and the picture should be symmetric about a vertical line through its middle.

Excessively long solutions that use more than 20 lines of code may lose points.

Answer:

int main() {
    int n;
    cout << "Enter a non-negative number n: ";
    cin >> n;
    if (n <= 0) return 0;

    for (int r = n - 1; r >= 0; r--) {
        for (int c = -(n - 1); c <= n - 1; c++)
            if (c < -r || c > r || (c + r) % 2 != 0) cout << " ";
            else cout << "*";
        cout << endl;
    }
    return 0;
}
for (int c = -(n - 1); c <= n - 1; c++)
    if (c < -r || c > r || (c + r) % 2 != 0) cout << " ";
    else cout << "X";
    cout << endl;
} return 0;
}

Alternative solution:

void triangle(int n, int indent) {
    if (n == 0) return;
    for (int c = 1; c <= indent; c++) cout << " ";
    for (int c = 1; c <= n; c++) cout << "X ";
    cout << endl;
    triangle(n - 1, indent + 1);
}
int main() {
    int n;
    cout << "Enter a non-negative number n: ";
    cin >> n;
    if (n <= 0) return 0;
    triangle(n, 0);
    return 0;
}

Problem 34 Write a program that uses a standard C++ function to generate a random integer secret between 1 and 10. It then asks the user to guess the secret number and gives at most 3 guesses. Your program should run to produce sample output as shown here:

You have 3 guesses to find my secret number.
What’s your next guess: 3
What’s your next guess: 4
Correct!

Excessively long solutions that use more than 20 lines of code may lose points.

Answer:

#include <cstdlib>

int main() {
    int secret = rand() % 10 + 1;
    cout << "You have 3 guesses to find my secret number.
";
    int guess, strikes = 0;
    while (strikes < 3) {
        cout << "What’s your next guess: ";
        cin >> guess;
        if (guess == secret) {
            cout << "Correct!\n";
            return 0;
        }
        strikes++;
    }
    cout << "You couldn’t guess it. My number was " << secret << endl;
Problem 35  Write a math tutor program that uses a standard C++ function to generate two random integers x and y between 6 and 12 (inclusive). Then ask the user to enter the product of the numbers and say whether the user’s answer is right or wrong. If the user is wrong give them a second chance to answer. Your program should produce output as shown here:

```
What’s the product of 10 and 8? 100
Wrong!
Try again: 80
Correct!

What’s the product of 10 and 8? 100
Wrong!
Try again: 90
Wrong again!
```

Excessively long solutions that use more than 20 lines of code may lose points.

Answer:

```cpp
#include <cstdlib>

int main() {
    int x = rand() % 7 + 6;
    int y = rand() % 7 + 6;
    cout << "What’s the product of " << x << " and " << y << "? ";
    int answer;
    cin >> answer;
    if (answer == x * y) cout << "Correct!\n";
    else {
        cout << "Wrong!\nTry again: ";
        cin >> answer;
        if (answer == x * y) cout << "Correct!\n";
        else cout << "Wrong again!\n";
    }
    return 0;
}
```

Problem 36  Write a program that reads a positive integer n from the user. If the user gives a non-positive value for n the program should terminate immediately. Otherwise, the program should print a triangle with n rows that looks like the following sample when n is 4.

```
*  
**X**
*X*X*X  
*X*X*X*  
```

The triangle should be made with * and X symbols with one extra of each per row. The *s should be separated by Xs and the picture should be symmetric about a vertical line through its middle.

Excessively long solutions that use more than 20 lines of code may lose points.

Answer:

```cpp
int main() {
    int n;
    cout << "Enter a non-negative number n: ";
    cin >> n;
    if (n <= 0) return 0;

    for (int r = 0; r < n; r++) {
        for (int c = -(n - 1); c <= n - 1; c++)
            if (c < -r || c > r) cout << " ";
        cout << "\n";
    }
    return 0;
}
```
else if (((c + r) % 2 != 0) cout << "X";
 else cout << "*";
 cout << endl;
}
 return 0;
}

Problem 37  The recursive function removeFirst removes the first digit of a positive integer. It is applied in another recursive function middle that returns the middle digit (or middle two digits) of a positive integer none of whose digits are 0. For example the middle of 12345 is 3 but the middle of 1234 is 23.
Implementations of these functions with parts of the code covered up are given below. There is also a main program that uses them.
Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the 5 parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

int removeFirst(PART (a)) {
 if (PART (b))
 return 0;
 PART (c)
}
int middle(int x) {
 if (PART (d))
 return x;
 PART (e)
}
int main() {
 cout << middle(19683) << endl; // prints 6
 cout << middle(1968) << endl; // prints 96
 return 0;
}

(a) Give a replacement for PART (a) to declare the parameter x
Answer: PART (a) is int x
(b) Give a replacement for PART (b) to test for the base case of recursion:
Answer: PART (b) is x < 10
(c) Give a replacement for PART (c) as a useful recursive call:
Answer: PART (c) is return 10 * removeFirst(x / 10) + x % 10;
(d) Give a replacement for PART (d) to test for the base case of recursion:
Answer: PART (d) is x < 100
(e) Give a replacement for PART (e) as a useful recursive call:
Answer: PART (e) is return middle(removeFirst(x) / 10);

Problem 38  The recursive function append uses two positive integer parameters x and y and returns an answer formed from the digits of x followed by those of y. For example, if the parameters are 196 and 83 the returned value is 19683.
The function is applied in another recursive function reverse that reverses a positive integer none of whose digits are 0. For example the reverse of 12345 is 54321.
Implementations of these functions with parts of the code covered up are given below. There is also a main program that uses them.
Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the 5 parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.
```cpp
int append(PART (a)) {
    if (PART (b))
        return x;
    PART (c)
}

int reverse(int x) {
    if (PART (d))
        return x;
    PART (e)
}

int main() {
    cout << reverse(19683) << endl; // prints 38691
    cout << reverse(1968) << endl; // prints 8691
    return 0;
}

(a) Give a replacement for PART (a) to declare parameters x and y
Answer: PART (a) is int x, int y

(b) Give a replacement for PART (b) to test for the base case of recursion:
Answer: PART (b) is y == 0

(c) Give a replacement for PART (c) as a useful recursive call:
Answer: PART (c) is return 10 * append(x, y / 10) + y % 10;

(d) Give a replacement for PART (d) to test for the base case of recursion:
Answer: PART (d) is x < 10

(e) Give a replacement for PART (e) as a useful recursive call:
Answer: PART (e) is return append(x % 10, reverse(x / 10));

Problem 39 The recursive function containsDigit uses two parameters. The first represents a positive integer and the second represents a digit (an integer between 0 and 9). The function returns a result of true if the first parameter has the second as one of its digits.

It is applied in another recursive function removeDuplicates that returns a result that keeps only the leftmost occurrence of each digit in a positive integer. For example removeDuplicates(1123243) would return 1234.

Implementations of these functions with parts of the code covered up are given below. There is also a main program that uses them.

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the 5 parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```
int main() {
    cout << removeDuplicates(1123243) << endl; // prints 1234
    cout << removeDuplicates(1143223) << endl; // prints 1432
    return 0;
}

(a) Give a replacement for PART (a) to test for the base case of recursion:
Answer: PART (a) is x == 0

(b) Give a replacement for PART (b) to test the last digit
Answer: PART (b) is x % 10 == d

(c) Give a replacement for PART (c) as a useful recursive call:
Answer: PART (c) is containsDigit(x / 10, d)

(d) Give a replacement for PART (d) as a useful recursive call:
Answer: PART (d) is removeDuplicates(x / 10)

(e) Give a replacement for PART (e) as another useful recursive call
Answer: PART (e) is 10 * removeDuplicates(x / 10) + x % 10

Problem 40  The recursive function removeDigit uses two parameters. The first represents a positive integer and the second represents a digit (an integer between 0 and 9). The function returns a result obtained by removing the digit from the first parameter. For example removeDigit(11231, 1) returns 23.
It is applied in another recursive function removeDuplicates that returns a result that keeps only the rightmost occurrence of each digit in a positive integer. For example removeDuplicates(1123243) would return 1243.
Implementations of these functions with parts of the code covered up are given below. There is also a main program that uses them.
Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the 5 parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

int removeDigit(int x, int d) {
    if (PART (a))
        return 0;
    if (x % 10 == d)
        return PART (b);
    return PART (c);
}

int removeDuplicates(int x) {
    if (x < 10) return x;
    int y = removeDuplicates(PART (d));
    return PART (e);
}

int main() {
    cout << removeDuplicates(1123243) << endl; // prints 1234
    cout << removeDuplicates(1143223) << endl; // prints 1432
    return 0;
}

(a) Give a replacement for PART (a) to test for base cases of recursion:
Answer: PART (a) is x == 0

(b) Give a replacement for PART (b) as a useful recursive call:
Answer: PART (b) is removeDigit(x / 10, d);

(c) Give a replacement for PART (c) to apply another useful recursive call:
Answer: PART (c) is removeDigit(x / 10, d) * 10 + x % 10
(d) Give a replacement for PART (d) as a useful recursive call
Answer: PART (d) is  \( \text{removeDigit}(x / 10, x \% 10) \)
(e) Give a replacement for PART (e) to compute an answer using \( x \) and \( y \)
Answer: PART (e) is  \( y \times 10 + x \% 10 \);

**Problem 41**  The recursive function removeFirst removes the first digit of a positive integer. It is applied in another recursive function middle that also has a positive integer parameter. The function middle returns either the middle digit if the parameter has an odd number of digits or the middle two digits if it has an even number of digits. For example the middle of 12345 is 3 but the middle of 1234 is 23. Implementations of these functions with parts of the code covered up are given below. There is also a main program that uses them. Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the 5 parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```cpp
int removeFirst(int x) {
    if (x < 10) return 0;
    PART (d)
}

int middle(int x) {
    if (x < 100) return x;
    PART (e)
}

int main() {
    cout << middle(19683) << endl; // prints 6
    cout << middle(1968) << endl; // prints 96
    return 0;
}
```

(a) Give a replacement for PART (a) to declare the parameter \( x \)
Answer: PART (a) is  \( \text{int } x \)
(b) Give a replacement for PART (b) to test for the base case of recursion:
Answer: PART (b) is  \( x < 10 \)
(c) Give a replacement for PART (c) the result for the base case
Answer: PART (c) is  \( 0 \)
(d) Give a replacement for PART (d) as a useful recursive call:
Answer: PART (d) is  \( \text{return } 10 \times \text{middle}(<x>_2) + x \% 10; \)
(e) Give a replacement for PART (e) as a useful recursive call:
Answer: PART (e) is  \( \text{return } \text{middle}(x / 10); \)

**Problem 42**  Write a function called adjust adds a random integer between -2 and 2 to each element of an array of integers. The biggest allowed change to an entry of the array is 2 (either up or down) and the smallest allowed change is 0. Excessively long solutions that use more than 10 lines of code may lose points.

Answer:

```cpp
int main() {
    int x[5] = {3,1,4,1,5};
    adjust(x, 5);
    cout << x[1] << endl; // prints a value that could be any of -1, 0, 1, 2, 3
    return 0;
}
```
Alternative solution:

```c
int main() {
    int x[5] = {3,1,4,1,5};
    adjust(x, 5);
    cout << x[1] << endl; // prints a value that could be any of -1, 0, 1, 2, 3
    return 0;
}
```

**Problem 43** Write a function called `peaks` that counts the number of elements in an array that are larger than both of their neighbors. (The first and last elements do not have 2 neighbors and are not to be counted.) Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function `peaks` follows.

```c
int main() {
    int x[8] = {3,1,4,1,5,9,2,6};
    cout << peaks(x, 8) << endl; // prints 2 since the elements 4 and 9 are counted
    return 0;
}
```

**Answer:**

```c
int peaks(int a[], int c) {
    int ans = 0;
    for (int i = 1; i < c - 1; i++)
        if (a[i] > a[i - 1] && a[i] > a[i + 1]) ans++;
    return ans;
}
```

**Problem 44** Write a function called `peaks` that prints the elements in an array that are larger than both of their neighbors. (The first and last elements do not have 2 neighbors and are not to be counted.) Different elements to be printed should be separated by spaces. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function `peaks` follows.

```c
int main() {
    int x[8] = {3,1,4,1,5,9,2,6};
    peaks(x, 8); // prints 4 9
    return 0;
}
```

**Answer:**

```c
void peaks(int a[], int c) {
    for (int i = 1; i < c - 1; i++)
        if (a[i] > a[i - 1] && a[i] > a[i + 1]) cout << a[i] << " ";
    cout << endl;
}
```
Problem 45  Write a function called *peaks* that prints the elements in an array that are larger than the average of the entries in the array. Different elements to be printed should be separated by spaces. Excessively long solutions that use more than 10 lines of code may lose points. A program that uses the function *peaks* follows.

```cpp
int main() {
    int x[8] = {3,1,4,1,5,9,2,6}; // has average 3.875
    peaks(x, 8); // prints 4 5 9 6
    return 0;
}
```

Answer:

```cpp
void peaks(int a[], int c) {
    int sum = 0;
    for (int i = 0; i < c ; i++)
        sum += a[i];
    double average = sum / (int) c;
    for (int i = 0; i < c; i++)
        if (a[i] > average) cout << a[i] << " ";
    cout << endl;
}
```

Problem 46  Write a function called *peaks* that returns the number of elements in an array that are larger than the average of the entries in the array. Excessively long solutions that use more than 10 lines of code may lose points. A program that uses the function *peaks* follows.

```cpp
int main() {
    int x[8] = {3,1,4,1,5,9,2,6}; // has average 3.875
    cout << peaks(x, 8) << endl; // prints 4 since the elements 4, 5, 9, 6 are above average
    return 0;
}
```

Answer:

```cpp
int peaks(int a[], int c) {
    int sum = 0, count = 0;
    for (int i = 0; i < c ; i++)
        sum += a[i];
    double average = sum / (int) c;
    for (int i = 0; i < c; i++)
        if (a[i] > average) count++;
    return count;
}
```

Problem 47  Write a function called *lows* that counts the number of elements in an array that are smaller than both of their neighbors. (The first and last elements do not have 2 neighbors and are not to be counted.) Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function *lows* follows.
int main() {
    int x[8] = {3,1,4,1,5,9,2,6};
    cout << lows(x, 8) << endl; // prints 3 since the elements 1,1 and 2 are counted
    return 0;
}

Answer:

int lows(int a[], int c) {
    int ans = 0;
    for (int i = 1; i < c - 1; i++)
        if (a[i] < a[i - 1] && a[i] < a[i + 1]) ans++;
    return ans;
}

Problem 48 Write a function called sixes that has a single parameter that is an integer greater than 99. It returns the integer obtained by changing each of the first 3 digits of the parameter to 6. Your function can return any result of your choice if it is given an illegal parameter value. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function sixes follows.

int main() {
    cout << sixes(6789) << endl; // prints 6669
    cout << sixes(123) << endl; // prints 666
    cout << sixes(12345) << endl; // prints 66645
    return 0;
}

Answer:

int sixes(int x) {
    if (x < 1000) return 666;
    return 10 * sixes(x / 10) + x % 10;
}

Problem 49 Write a function called duplicates that has a single parameter that is a positive integer. It returns the integer obtained by changing all digits in the parameter to the first digit. Your function can return any result of your choice if it is given an illegal parameter value. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function duplicates follows.

int main() {
    cout << duplicates(6789) << endl; // prints 6666
    cout << duplicates(123) << endl; // prints 111
    cout << duplicates(12345) << endl; // prints 11111
    return 0;
}

Answer:
```c
int duplicates(int x) {
    if (x < 10) return x;
    int y = duplicates(x / 10);
    return 10 * y + y % 10;
}

Problem 50  Write a function called sixes that has a single parameter that is a positive integer. It returns the integer obtained by changing all digits of the parameter to 6.
Your function can return any result of your choice if it is given an illegal parameter value. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function sixes follows.

```c
int main() {
    cout << sixes(6789) << endl; // prints 6666
    cout << sixes(123) << endl; // prints 666
    cout << sixes(12345) << endl; // prints 66666
    return 0;
}
```

Answer:

```c
int sixes(int x) {
    if (x < 10) return 6;
    return 10 * sixes(x / 10) + 6;
}
```

Problem 51  Write a function called duplicates that has a single parameter that is an integer greater than 99. It returns the integer obtained by changing each of the first 3 digits in the parameter to the first digit.
Your function can return any result of your choice if it is given an illegal parameter value. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function duplicates follows.

```c
int main() {
    cout << duplicates(6789) << endl; // prints 6669
    cout << duplicates(123) << endl; // prints 111
    cout << duplicates(12345) << endl; // prints 11145
    return 0;
}
```

Answer:

```c
int duplicates(int x) {
    if (x < 1000) return x/100 * 111;
    return 10 * duplicates(x / 10) + x % 10;
}
```

Problem 52  Write a function called twos that has a single parameter that is an integer greater than 9. It returns the integer obtained by changing each of the first 2 digits of the parameter to 2.
Your function can return any result of your choice if it is given an illegal parameter value. Excessively long solutions that use more than 8 lines of code may lose points. A program that uses the function twos follows.

```c
int main() {
    cout << twos(6789) << endl; // prints 6666
    cout << twos(123) << endl; // prints 111
    cout << twos(12345) << endl; // prints 11145
    return 0;
}
```

Answer:

```c
int twos(int x) {
    if (x < 1000) return x/100 * 111;
    return 10 * twos(x / 10) + x % 10;
}
```
int main() {
    cout << twos(6789) << endl; // prints 2289
    cout << twos(123) << endl;  // prints 223
    cout << twos(12345) << endl; // prints 22345
    cout << twos(13) << endl;   // prints 22
    return 0;
}

Answer:

int twos(int x) {
    if (x < 100) return 22;
    return 10 * twos(x / 10) + x % 10;
}