

Solutions

08.30am – 10.30am, Monday, December 18, 2017

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

```
int main() {  
    int x = 0, y = 1, z = 2;  
    double w[3] = {1.9, 2.3, 3.0};  
  
    x = a(x + y, z);           // (a) sets x as the smaller of two values  
    w[0] = b(x, y, y, w[2]);   // (b) sets w[0] as the largest of four values  
    c(w, y, x);               // (c) print the values of w indexed by x and y  
    d(w[1], y);               // (d) increase y by the nearest integer to w[1]  
    d(e(y, z), y);            // (e) applies e and then d  
    return 0;  
}
```

(a) Title line for **a**.

Answer:

```
int a(int x, int y)
```

(b) Title line for **b**.

Answer:

```
double b(int x, int y, int z, double w)
```

(c) Title line for **c**.

Answer:

```
void c(double w[], int x, int y)
```

(d) Title line for **d**.

Answer:

```
void d(double x, int &y)
```

(e) Title line for **e**.

Answer:

```
double e(int x, int y)
```

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;

int fun(int x) {
    int ans = 23456;
    if (x <= 0) return -1;
    if ((x >= 5) && (x < 10)) return ans % 1000;
    if (x >= 7) return -2;
    cout << x / 2;
    return fun(x + 1);
}

int main() {
    cout << fun(0) << endl;    // line (a)
    cout << fun(6) << endl;    // line (b)
    cout << fun(7) << endl;    // line (c)
    cout << fun(17) << endl;   // line (d)
    cout << fun(3) << endl;    // line (e)
}
```

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

Answer:

-1

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

Answer:

456

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

Answer:

456

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

Answer:

-2

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

Answer:

12456

Problem 3 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 of only a few lines.

```
int main() {
    int i = 2;
    int x[5] = {3, 1, 4, 1, 5};
    // (a) Return the absolute value (ignoring a minus sign). Here 2 is printed.
    cout << absVal(-2) << endl;
    // (b) Return number of even entries, here 1 is printed.
    cout << numEven(x, 5) << endl;
    // (c) Cube i. Here 8 is printed.
    cubeIt(i); cout << i << endl;
    // (d) Find the (last) index of the smallest entry. Here 3 is printed.
    cout << findIndexMin(x, 5) << endl;
    // (e) Is it a digit? Here print nothing.
    if (isDigit('h')) cout << "Digit" << endl;
    return 0;
}
```

Answer:

(a)

```
int absVal(int x) {
    if (x < 0) return -x;
    return x;
}
```

(b)

```
int numEven(int array[], int cap) {
    int ans = 0;
    for (int i = 0; i < cap; i++)
        if (array[i] % 2 == 0) ans++;
    return ans;
}
```

(c)

```
void cubeIt(int &x) {
    x = x * x * x;
}
```

(d)

```
int findIndexMin(int array[], int cap) {
    int ans = 0;
    for (int i = 0; i < cap; i++)
        if (array[i] <= array[ans]) ans = i;
    return ans;
}
```

(e)

```
bool isDigit(char x) {
    return '0' <= x && x <= '9';
}
```

Problem 4 Write a complete C++ program that does the following:

1. Asks the user to enter 2 positive integers, x and y . If either is illegal then the program terminates.
2. Prints all integers n with $x \leq n \leq x^2$ for which the sum of the digits of n is exactly equal to y .

The numbers printed should appear on separate lines of output. Excessively long solutions (with more than 25 lines of code) may lose points.

For example, the following represents one run of the program:

```
Enter 2 integers : 5 6
6
15
24
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
    int x, y;
    cout << "Enter 2 integers : ";
    cin >> x >> y;
    if (x <= 0 || y <= 0) return 0;
    for (int n = x; n <= x * x; n++) {
        int sum = 0, copy = n;
        while (copy > 0) {
            sum += copy % 10;
            copy /= 10;
        }
        if (sum == y) cout << n << endl;
    }
    return 0;
}
```

Problem 5 Write a function called *firstDuplicate* that reports the first duplicate that it finds in an array of characters. If there is no duplicate your function should return '?' as its answer. Your solution should use no more than 15 lines of code.

For example, a program that uses the function *firstDuplicate* follows.

```
int main() {
    char x[7] = {'Q', 'u', 'e', 'e', 'n', 's', 'Q'};
    cout << firstDuplicate(x, 7) << endl;    // prints    e
    return 0;
}
```

In this example, the second letter *e* is the first duplicate found in the array. The duplicate letter *Q* comes later.

Answer:

```
char firstDuplicate(char x[], int c) {
    for (int i = 1; i < c; i++)
        for (int j = 0; j < i; j++)
            if (x[i] == x[j]) return x[i];
    return '?';
}
```

Problem 6 Write a function called *biggerDigits* that uses two positive integer parameters with the same number of digits and returns a result of *true* if every digit in the first parameter is bigger than the corresponding digit in the second parameter. Otherwise it returns *false*. If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 6 lines of code may lose points.

For example, a program that uses the function *biggerDigits* follows.

```
int main() {
    cout << biggerDigits(987, 123) << endl;           // prints true
    cout << biggerDigits(123, 987) << endl;           // prints false
    cout << biggerDigits(98765, 12345) << endl;       // prints false
                                                // because the last digit isn't bigger
    if (biggerDigits(76, 91)) cout << "Hello";       // doesn't print
    return 0;
}
```

Answer:

```
bool biggerDigits(int x, int y) {
    if (x == 0) return true;
    if ((x % 10) <= (y % 10)) return false;
    return biggerDigits(x/10, y/10);
}
```

Solutions

08.30am – 10.30am, Monday, December 18, 2017

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

```
int main() {
    double x = 0, y = 1, z = 2;
    double w[3] = {1.9, 2.3, 3.0};

    x = f1(x + y, z);           // (a) sets x as the smaller of two values
    w[0] = f2(x, y, y, w[2]);   // (b) sets w[0] as the largest of four values
    f3(w, 3);                   // (c) print all values in w
    f4(w[1], y);                // (d) decrease y by w[1]
    f4(f5(y, z), y);           // (e) applies f5 and then f4
    return 0;
}
```

(a) Title line for **f1**.

Answer:

```
double f1(double x, double y)
```

(b) Title line for **f2**.

Answer:

```
double f2(double x, double y, double z, double w)
```

(c) Title line for **f3**.

Answer:

```
void f3(double w[], int c)
```

(d) Title line for **f4**.

Answer:

```
void f4(double x, double &y)
```

(e) Title line for **f5**.

Answer:

```
double f5(double x, double y)
```

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;

int fun(int x) {
    int ans = 2345;
    if (x <= 0) return -2;
    if ((x >= 6) && (x < 10)) return ans % 100;
    if (x >= 8) return -5;
    cout << x / 3;
    return fun(x - 1);
}

int main() {
    cout << fun(0) << endl;    // line (a)
    cout << fun(6) << endl;    // line (b)
    cout << fun(7) << endl;    // line (c)
    cout << fun(17) << endl;   // line (d)
    cout << fun(3) << endl;    // line (e)
}
```

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

Answer:

-2

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

Answer:

45

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

Answer:

45

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

Answer:

-5

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

Answer:

100-2

Problem 3 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 of only a few lines.

```
int main() {
    int i = 2;
    int x[7] = {3, 1, 4, 1, 5, 9, 2};
    // (a) Return the exact quotient. Here 0.4 is printed.
    cout << divide(i, 5) << endl;
    // (b) Return number of odd entries. Here 5 is printed.
    cout << numOdd(x, 7) << endl;
    // (c) Make a number from two copies of a (single) digit. Here 22 is printed.
    cout << doubleIt(2) << endl;
    // (d) Find the last index of the largest entry. Here 5 is printed.
    cout << findIndexMax(x, 7) << endl;
    // (e) Is it a lower case character? Here L is printed.
    if (isLowerCase('h')) cout << "L" << endl;
    return 0;
}
```

Answer:

(a)

```
double divide(int x, int y) {
    return x / ((double) y);
}
```

(b)

```
int numOdd(int array[], int cap) {
    int ans = 0;
    for (int i = 0; i < cap; i++)
        if (array[i] % 2 != 0) ans++;
    return ans;
}
```

(c)

```
int doubleIt(int x) {
    return 11 * x;
}
```

(d)

```
int findIndexMax(int array[], int cap) {
    int ans = 0;
    for (int i = 0; i < cap; i++)
        if (array[i] >= array[ans]) ans = i;
    return ans;
}
```

(e)

```
bool isLowerCase(char x) {
    return 'a' <= x && x <= 'z';
}
```

Problem 4 Write a complete C++ program that does the following:

1. Asks the user to enter 2 positive integers, x and y . If either is illegal then the program repeatedly asks the user to retype x and y until legal values are given.
2. Prints all integers n with $1 \leq n \leq x$ for which the product of the digits of n is exactly equal to y .

The numbers printed should appear on separate lines of output. Excessively long solutions (with more than 25 lines of code) may lose points.

For example, the following represents one run of the program:

```
Enter 2 integers : 35 6
6
16
23
32
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
    int x, y;
    cout << "Enter 2 integers : ";
    cin >> x >> y;
    while (x <= 0 || y <= 0) {
        cout << "Illegal. Enter 2 positive integers: ";
        cin >> x >> y;
    }
    for (int n = 1; n <= x; n++) {
        int product = 1, copy = n;
        while (copy > 0) {
            product *= copy % 10;
            copy /= 10;
        }
        if (product == y) cout << n << endl;
    }
    return 0;
}
```

Problem 5 Write a function called *firstUnique* that reports the first entry that has no duplicate in an array of integers. If there is no such entry your function should return -1 as its answer. Your solution should use no more than 15 lines of code.

For example, a program that uses the function *firstUnique* follows.

```
int main() {
    int x[10] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3};
    cout << firstUnique(x, 10) << endl;    // prints 4
    return 0;
}
```

In this example, first two entries of 3 and 1 have later duplicates, so the result is given by the third entry of 4.

Answer:

```
int firstUnique(int x[], int c) {
    for (int i = 0; i < c; i++) {
        int count = 0;
        for (int j = 0; j < c; j++)
            if (x[i] == x[j]) count++;
        if (count == 1) return x[i];
    }
    return -1;
}
```

Problem 6 Write a function called *sameEvens* that uses two positive integer parameters with the same number of digits and returns a result of *true* if the positions of the even digits in the two parameters are identical. Otherwise it returns *false*. For example, the even digits in both of the numbers 12345 and 98765 occupy the 2nd and 4th positions so that `sameEvens(12345, 98765)` would return *true*.

If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 6 lines of code may lose points.

For example, a program that uses the function *sameEvens* follows.

```
int main() {
    cout << sameEvens(987, 123) << endl;           // prints true
    cout << sameEvens(123, 223) << endl;           // prints false
    cout << sameEvens(98765, 12345) << endl;       // prints true
    if (sameEvens(76, 91)) cout << "Hello";       // doesn't print
    return 0;
}
```

Answer:

```
bool sameEvens(int x, int y) {
    if (x == 0) return true;
    if ((x % 2) != (y % 2)) return false;
    return sameEvens(x/10, y/10);
}
```

Solutions

01.45pm – 03.45pm, Monday, December 18, 2017

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

```
int main() {  
    char x = '0', y = '1', z = '2';  
    string w[3] = {"1.9", "2.3", "3.0"};  
  
    x = a(x, z);           // (a) sets x as the smaller of two characters  
    w[0] = b(x, y, y, w[2]); // (b) sets w[0] as the concatenation  
    c(w, 0, 1);           // (c) prints the concatenation of w[0] and w[1]  
    d(w[1], y);           // (d) change y to the first character of w[1]  
    d(e(y, z), y);        // (e) applies e and then d  
    return 0;  
}
```

(a) Title line for **a**.

Answer:

```
char a(char x, char y)
```

(b) Title line for **b**.

Answer:

```
string b(char x, char y, char z, string w)
```

(c) Title line for **c**.

Answer:

```
void c(string w[], int x, int y)
```

(d) Title line for **d**.

Answer:

```
void d(string x, char &y)
```

(e) Title line for **e**.

Answer:

```
string e(char x, char y)
```

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;

int fun(int x) {
    int ans = 34567;
    if (x <= 0) return 0;
    if ((x >= 6) && (x < 10)) return ans % 1000;
    if (x >= 8) return -1;
    cout << x % 2;
    return fun(x + 2);
}

int main() {
    cout << fun(0) << endl;    // line (a)
    cout << fun(6) << endl;    // line (b)
    cout << fun(7) << endl;    // line (c)
    cout << fun(17) << endl;   // line (d)
    cout << fun(3) << endl;    // line (e)
}
```

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

Answer:

0

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

Answer:

567

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

Answer:

567

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

Answer:

-1

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

Answer:

11567

Problem 3 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 of only a few lines.

```
int main() {
    int a = 123, b = 3;
    ifstream f;
    string s = "HELLO";
    f.open("testFile.txt");
    // (a) Tests whether a number has 3 digits, here Yes!
    if (is3digit(a)) cout << "Yes!" << endl;
    // (b) Doubles a string, here HELLOHELLO
    cout << doubleIt(s) << endl;
    // (c) Returns the number of words found in the input file before eof() is true
    cout << countWords(f) << endl;
    // (d) Print middle character of a string that has odd length here L, ignore even lengths
    cout << midChar(s) << endl;
    // (e) swap a and b so that 3,123 is printed
    swap(a, b);
    cout << a << "," << b << endl;
    return 0;
}
```

Answer:

(a)

```
bool is3digit(int x) {
    return (x > 99) && (x < 1000);
}
```

(b)

```
string doubleIt(string x) {
    return x + x;
}
```

(c)

```
int countWords(ifstream &file) {
    string x; int count = 0;
    while (!file.eof()) {
        file >> x; count++;
    }
    return count;
}
```

(d)

```
char midChar(string x) {
    return x[x.length()/2];
}
```

(e)

```
void swap(int &x, int &y) {
    int temp = x;
    x = y; y = temp;
}
```

Problem 4 Write a complete C++ program that does the following:

1. Asks the user to enter 2 positive integers, x and y for which $0 < y \leq 9$. If either is illegal then the program terminates.
2. Prints all integers n with $x \leq n < x^2$ such that one of the digits of n is equal to y .

The numbers printed should appear on separate lines of output. Excessively long solutions (with more than 25 lines of code) may lose points.

For example, the following represents one run of the program:

```
Enter 2 integers : 5 2
12
20
21
22
23
24
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
    int x, y;
    cout << "Enter 2 integers : ";
    cin >> x >> y;
    if (x <= 0 || y <= 0 || y > 9) return 0;
    for (int n = x; n < x * x; n++) {
        int count = 0, copy = n;
        while (copy > 0) {
            if (copy % 10 == y) count++;
            copy /= 10;
        }
        if (count > 0) cout << n << endl;
    }
    return 0;
}
```


Problem 5 Write a function called *firstUniqueIndex* that reports the index of the first entry that has no duplicate in an array of integers. If there is no such entry your function should return **-1** as its answer. Your solution should use no more than 15 lines of code.

For example, a program that uses the function *firstUniqueIndex* follows.

```
int main() {
    int x[10] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3};
    cout << firstUniqueIndex(x, 10) << endl;    // prints    2
    return 0;
}
```

In this example, indices 0 and 1 give entries of 3 and 1 that have later duplicates, so the result is the index 2.

Answer:

```
int firstUniqueIndex(int x[], int c) {
    for (int i = 0; i < c; i++) {
        int count = 0;
        for (int j = 0; j < c; j++)
            if (x[i] == x[j]) count++;
        if (count == 1) return i;
    }
    return -1;
}
```

Problem 6 Write a function called *sumGaps* that uses two positive integer parameters with the same number of digits and returns the sum of the gaps between their corresponding digits. For example if the numbers are 646 and 920 the gaps between their digits are 3 (between 6 and 9), 2 (between 4 and 2) and 6 (between 6 and 0).

If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 6 lines of code may lose points.

For example, a program that uses the function *sumGaps* follows.

```
int main() {
    cout << sumGaps(9, 1) << endl;           // prints 8
    cout << sumGaps(123, 987) << endl;       // prints 18
    cout << sumGaps(91, 19) << endl;        // prints 16
    return 0;
}
```

Answer:

```
int sumGaps(int x, int y) {
    if (x == 0) return 0;
    if ((x % 10) > (y % 10)) return x % 10 - y % 10 + sumGaps(x/ 10, y/10);
    return y% 10 - x % 10 + sumGaps(x/10, y/10);
}
```

Solutions

01.45pm – 03.45pm, Monday, December 18, 2017

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

```
int main() {
    double x = 0, y = 1, z = 2;
    string w[3] = {"1.9", "2.3", "3.0"};

    x = f1(x + y, z);           // (a) sets x as the smaller of two values
    w[0] = f2(x, y, y, w[2]);   // (b) sets w[0] using the four values
    f3(w, 3);                   // (c) print all values in w
    f4(w[1], y);                 // (d) decrease y by the numerical value of w[1]
    f4(f5(y, z), y);            // (e) applies f5 and then f4
    return 0;
}
```

(a) Title line for **f1**.

Answer:

```
double f1(double x, double y)
```

(b) Title line for **f2**.

Answer:

```
string f2(double x, double y, double z, string w)
```

(c) Title line for **f3**.

Answer:

```
void f3(string w[], int c)
```

(d) Title line for **f4**.

Answer:

```
void f4(string x, double &y)
```

(e) Title line for **f5**.

Answer:

```
string f5(double x, double y)
```

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;

int fun(int x) {
    int ans = 5432;
    if (x <= 0) return -1;
    if ((x >= 8) && (x < 13)) return ans % 100;
    if (x >= 10) return -5;
    cout << x % 3;
    return fun(x + 2);
}

int main() {
    cout << fun(0) << endl;    // line (a)
    cout << fun(6) << endl;    // line (b)
    cout << fun(7) << endl;    // line (c)
    cout << fun(17) << endl;   // line (d)
    cout << fun(3) << endl;    // line (e)
}
```

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

Answer:

-1

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

Answer:

032

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

Answer:

132

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

Answer:

-5

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

Answer:

02132

Problem 3 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 of only a few lines.

```
int main() {
    int a = 2, b = 3, c = 4;
    ifstream f;
    string s = "HELLO";
    f.open("testFile.txt");
    // (a) Tests whether a number is even, here Even!
    if (even(c)) cout << "Even!" << endl;
    // (b) Removes first character from a string, here ELL0
    cout << removeFirst(s) << endl;
    // (c) Returns first word read from the input file
    cout << firstWord(f) << endl;
    // (d) Returns last character of a string, here 0
    cout << lastChar(s) << endl;
    // (e) Change a,b,c to be c, a, b so here it prints 423
    rotate(a, b, c);
    cout << a << b << c << endl;
    return 0;
}
```

Answer:

(a)

```
bool even(int x) {
    return x % 2 == 0;
}
```

(b)

```
string removeFirst(string x) {
    return x.substr(1);
}
```

(c)

```
string firstWord(ifstream &file) {
    string x;
    file >> x;
    return x;
}
```

(d)

```
char lastChar(string x) {
    return x[x.length() - 1];
}
```

(e)

```
void rotate(int &x, int &y, int &z) {
    int temp = x;
    x = z;
    z = y;
    y = temp;
}
```

Problem 4 Write a complete C++ program that does the following:

1. Asks the user to enter 2 positive integers, x and y for which $0 < y \leq 9$. If either is illegal then the program should repeatedly ask the user to re-enter x and y .
2. Prints all integers n with $1 \leq n \leq x$ such that all of the digits of n are at least as large as y .

The numbers printed should appear on separate lines of output. Excessively long solutions (with more than 25 lines of code) may lose points.

For example, the following represents one run of the program:

```
Enter 2 integers : 100 8
8
9
88
89
98
99
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
    int x, y;
    cout << "Enter 2 integers : ";
    cin >> x >> y;
    while (x <= 0 || y <= 0 || y > 9) {
        cout << "Illegal. try again: ";
        cin >> x >> y;
    }
    for (int n = 1; n <= x; n++) {
        int copy = n;
        bool ok = true;
        while (copy > 0) {
            if (copy % 10 < y) ok = false;
            copy /= 10;
        }
        if (ok) cout << n << endl;
    }
    return 0;
}
```

Problem 5 Write a function called *firstDuplicateIndex* that reports the first index that contains a duplicate of an earlier entry in an array of characters. If there is no duplicate your function should return -1 as its answer. Your solution should use no more than 15 lines of code.

For example, a program that uses the function *firstDuplicateIndex* follows.

```
int main() {
    char x[7] = {'Q', 'u', 'e', 'e', 'n', 's', 'Q'};
    cout << firstDuplicateIndex(x, 7) << endl;    // prints 3
    return 0;
}
```

In this example, the letter at index *e* which duplicates the earlier *e* at index 2.

Answer:

```
int firstDuplicateIndex(char x[], int c) {
    for (int i = 1; i < c; i++)
        for (int j = 0; j < i; j++)
            if (x[i] == x[j]) return i;
    return -1;
}
```

Problem 6 Write a function called *productGaps* that uses two positive integer parameters with the same number of digits and returns the product of the gaps between their corresponding digits. For example if the numbers are 646 and 920 the gaps between their digits are 3 (between 6 and 9), 2 (between 4 and 2) and 6 (between 6 and 0).

If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 6 lines of code may lose points.

For example, a program that uses the function *productGaps* follows.

```
int main() {
    cout << productGaps(9, 1) << endl;           // prints 8
    cout << productGaps(678, 987) << endl;       // prints 3
    cout << productGaps(91, 19) << endl;        // prints 64
    return 0;
}
```

Answer:

```
int productGaps(int x, int y) {
    if (x == 0) return 1;
    if ((x % 10) > (y % 10)) return (x % 10 - y % 10) * productGaps(x/ 10, y/10);
    return (y% 10 - x % 10) * productGaps(x/10, y/10);
}
```