

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 b[3] = {1.9, 2.3, 3.0};
    int d[2][2] = {{1,2},{3,4}};

    x = sum(z, y);                // (a) sets x to the sum: 3
    reset(d[1][1], z);            // (b) replaces d[1][1] by the value of z
    diagonal(d, 2, 2);            // (c) prints diagonal: 1 4
    cout << printAll(d, 2, 2) << endl; // (d) prints array: 1 2 3 4
    cout << add(b[2], d[0][0]) << endl; // (e) prints the sum: 4
    return 0;
}
```

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

Answer:

```
int sum(int z, int y)
```

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

Answer:

```
void reset(int &x, int y)
```

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

Answer:

```
void diagonal(int d[][2], int r, int c)
```

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

Answer:

```
string printAll(int d[][2], int r, int c)
```

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

Answer:

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

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;
int recursive (int x) {
    if (x < 5) return 3;
    return recursive (x / 3) + x % 6;
}

char swap (int x, int y) {
    x = y;
    y = x;
    cout << x << y;
    return 's';
}

void set (int arr []) {
    arr[0] = 1; arr[1] = 9; arr[2] = 6; arr[3] = 8; arr[4] = 3;
}

int main() {
    int x[5];
    set(x);
    swap(1, 2); //line (a)
    set(x);
    cout << x[0 + 2] << x[0] + 2 << endl; //line (b)
    cout << swap(1, 2) << endl; //line (c)
    for (int i = 1; i < 4; i++) cout << x[i]; cout << endl; //line (d)
    int e = 21;
    cout << recursive(e) << endl; //line (e)
    return 0;
}
```

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

Answer:

22

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

Answer:

63

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

Answer:

22s

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

Answer:

968

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

Answer:

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 sum. Here 4 is printed.
    cout << add(i, 2) << endl;
    // (b) Return number of odd entries. Here 4 is printed.
    cout << numOdd(x, 5) << endl;
    // (c) Multiply i by 2. Here 4 is printed.
    doubleIt(i); cout << i << endl;
    // (d) Find the index of the largest entry. Here 4 is printed.
    cout << findIndexMax(x, 5) << endl;
    // (e) Is it a lower case character? Here 4 is printed.
    if (isLowerCase('h')) cout << "4" << endl;
    return 0;
}
```

Answer:

(a)

```
int add(int x, int y) {
    return x + 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)

```
void doubleIt(int &x) {
    x *= 2;
}
```

(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 function called *noEl* that returns the number of elements that do not contain the letter *l* in a 2-dimensional array of strings (that has 3 columns).

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

```
int main() {
    string x[2][3] = {"CSCI", "One", "eleven"}, {"Queens", "College", "CUNY"};
    cout << noEl(x, 2, 3) << endl;    // prints: 4
    return 0;
}
```

Answer:

```
int noEl(string data[][3], int rows, int cols) {
    int count = 0;
    for (int r = 0; r < rows; r++)
        for (int c = 0; c < cols; c++)
            if ((int) data[r][c].find("l") < 0) count++;
    return count;
}
```

Problem 5 Write a function called *removeDuplicates* that replaces any sequence of copies of a digit in a positive integer parameter by a single copy of that digit.

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

```
int main() {
    cout << removeDuplicates(55511) << endl;           // prints 51
    cout << removeDuplicates(51155) << endl;           // prints 515
    cout << removeDuplicates(551155) << endl;          // prints 515
    cout << removeDuplicates(515) << endl;             // prints 515
    return 0;
}
```

Answer:

```
int removeDuplicates(int x) {
    if (x < 10) return x;
    int y = removeDuplicates(x / 10);
    if (y % 10 == x % 10) return y;
    return 10 * y + x % 10;
}
```

Problem 6 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter 25 integers and it reads the numbers that the user gives.
2. It calculates the average of the entered numbers.
3. It reports all entered numbers that are greater than the average, by printing them to a file called output6.txt.

Answer:

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

int main() {
    ofstream out;
    out.open("output6.txt");
    int x[25];
    cout << " Enter 25 integers: ";
    for (int i = 0; i < 25; i++) cin >> x[i];
    int sum = 0;
    for (int i = 0; i < 25; i++) sum += x[i];
    double average = sum / 25.0;
    for (int i = 0; i < 25; i++)
        if (x[i] > average) out << x[i] << endl;
    out.close();
    return 0;
}
```

Solutions

11.00am – 01.00pm, Tuesday, May 19, 2015

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 d = 2;
    string x[5] = {"3", "1", "4", "1", "5"};
    d = average(x, 5);           // (a) sets d to 2.8
    d = max(d, x[4], 3); cout << d << endl; // (b) prints 5.0
    cout << inWords(x[1]) << endl;         // (c) prints one
    cout << f(f(x[0],d), 1.0) << endl;     // (d) mystery function prints 1.0
    percentage(8.0, x[2]);              // (e) prints 200%
    return 0;
}
```

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

Answer:

```
double average(string y[], int cap)
```

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

Answer:

```
double max(double x, string y, int z)
```

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

Answer:

```
string inWords(string x)
```

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

Answer:

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

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

Answer:

```
void percentage(double x, string y)
```

Problem 2 Consider the following C++ program.

```
#include <iostream>
using namespace std;
int recursive (int x) {
    if (x < 5) return 4;
    return recursive (x / 4) + x % 6;
}

char swap (int x, int y) {
    y = x;
    x = y;
    cout << x << y;
    return '0';
}

void set (int arr []) {
    arr[0] = 5; arr[1] = 9; arr[2] = 0; arr[3] = 4; arr[4] = 9;
}

int main() {
    int x[5];
    set(x);
    swap(1, 2); //line (a)
    set(x);
    cout << x[0 + 2] << x[0] + 2 << endl; //line (b)
    cout << swap(1, 2) << endl; //line (c)
    for (int i = 1; i < 4; i++) cout << x[i]; cout << endl; //line (d)
    int e = 21;
    cout << recursive(e) << endl; //line (e)
    return 0;
}
```

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

Answer:

11

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

Answer:

07

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

Answer:

110

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

Answer:

904

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

Answer:

12

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 absoluteValue. Here 2 is printed.
    cout << absoluteValue(i) << 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 (first) index of the smallest entry. Here 1 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 absoluteValue(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 function called *cString* that returns a comma separated list of all elements that start with the letter *C* in an array of strings.

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

```
int main() {
    string x[6] = {"Computer", "Science", "111", "Queens", "College", "CUNY"};
    cout << cString(x, 6) << endl;    // prints:  Computer,College,CUNY
    return 0;
}
```

Answer:

```
string cString(string data[], int cap) {
    string ans = "";
    for (int c = 0; c < cap; c++)
        if (data[c].find("C") == 0) {
            if (ans != "") ans = ans + ",";
            ans = ans + data[c];
        }
    return ans;
}
```

Problem 5 Write a function called *makeDecreasing* that makes a result with decreasing digits from a positive integer parameter. It selects the leftmost digit of the parameter and then later digits that are smaller than all that have already been selected.

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

```
int main() {
    cout << makeDecreasing(89321) << endl;           // prints 8321
    cout << makeDecreasing(892321) << endl;           // prints 821
    cout << makeDecreasing(1995) << endl;             // prints 1
    cout << makeDecreasing(7) << endl;                // prints 7
    return 0;
}
```

Answer:

```
int makeDecreasing(int x) {
    if (x < 10) return x;
    int y = makeDecreasing(x / 10);
    if (y % 10 <= x % 10) return y;
    return 10 * y + x % 10;
}
```

Problem 6 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter 25 integers and it reads the numbers that the user gives.
2. It calculates the smallest of the entered numbers.
3. It reports all entered numbers that are greater than the square of the smallest one. This output is to be printed to a file called output6.txt (and not to the user's screen).

Answer:

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

int main() {
    ofstream out;
    out.open("output6.txt");
    int x[25];
    cout << " Enter 25 integers: ";
    for (int i = 0; i < 25; i++) cin >> x[i];
    int smallest = x[0];
    for (int i = 0; i < 25; i++)
        if (x[i] < smallest) smallest = x[i];
    for (int i = 0; i < 25; i++)
        if (x[i] > smallest * smallest) out << x[i] << endl;
    out.close();
    return 0;
}
```