QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version A Exam Fall 2014 12.23.14

Solutions

08.30am – 10.30am, Tuesday, December 23, 2014

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 a[3] = {1, 1, 1}, i = 7, j = 8, k = 9;
   int b[5] = {1, 9, 6, 8, 3};
   int x[2][2] = {{2, 0}, {4, 8}};
   cout << max(i, j, k) << endl;   // (a) prints: 9
   printMax(b, 5);                 // (b) prints: 9
   cout << max2d(x, 2, 2) << endl; // (c) prints: 8
   swap (i, j);                    // (d) swaps i and j
   swapArrays (a, b, 2); // (e) swaps first 2 elements of arrays a and b
   return 0;
}
```

(a) Title line for **max**.
Answer:

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

(b) Title line for **printMax**.
Answer:

```
void printMax(int x[], int capacity)
```

(c) Title line for **max2d**.
Answer:

```
int max2d(int x[][2], int r, int c)
```

(d) Title line for **swap**.
Answer:

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

(e) Title line for **swapArrays**.
Answer:

```
void swapArrays(int x[], int y[], int number)
```

Problem 2 Consider the following C++ program.

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

void yesNo(bool ans) {
   if (ans) cout << "Y";
   else cout << "N";
   cout << endl;
}

int main() {
    int x = 3, y = 4, z = 5, a[4] = {0, 1, 2, 3};
    if (x == y) cout << "Y\n"; else cout << "N\n";         // line (a)
    if (x == a[x]) cout << "Y\n"; else cout << "N\n";      // line (b)
    if (!(x != y)) cout << "Y\n"; else cout << "N\n";      // line (c)
    yesNo((y < z) && (z < x));                             // line (d)
    yesNo((x < y) || (z < y));                             // line (e)
}
```

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

Answer:

```
N
```

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

Answer:

```
Y
```

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

Answer:

```
N
```

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

Answer:

```
N
```

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

Answer:

```
Y
```

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() {
      double a[4] = {1.0, 2.0, -3.0, -4.0};
      double b[4] = {0.5, 1.5, 2.5, 3.5};
// (a) Return the absolute value (ignoring sign).  Here 7 is printed.
      cout << absoluteValue(-7) << endl;
// (b) Return x/2 if x is even, otherwise 3*x+1: Here 22 is printed.
      cout << collatz(7) << endl;
// (c) Return the least factor. (Assume input at least 2.)  Here 5 is printed.
      cout << leastFactor(35) << endl;
// (d) Test whether all array entries are positive.  Here: Not all positive
      if (!allPositive(a, 4)) cout << "Not all positive\n";
// (e) Swap entries of the two arrays.
      swapArrays(a, b, 4);
      return 0;
}
```

Answer:

(a)

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

(b)

```
int collatz(int x) {
    if (x % 2 == 0) return x / 2;
    return 3 * x + 1;
}
```

(c)

```
int leastFactor(int x) {
    int ans = 2;
    while (x % ans != 0) ans++;
    return ans;
}
```

(d)

```
bool allPositive(double x[], int capacity) {
   for (int i = 0; i < capacity; i++)
      if (x[i] <= 0) return false;
   return true;
}
```

(e)

```
void swapArrays(double x[], double y[], int capacity) {
   for (int i = 0; i < capacity; i++) {
      double temp = x[i];
      x[i] = y[i];
      y[i] = temp;
   }
}
```

Problem 4 Write a function called *longestString* that returns the longest element in a 2-dimensional array of strings (that is known to have 2 columns).

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

```
int main() {
   string x[3][2] = {{"This", "is"}, {"an", "easy"}, {"question", ""}} ;
   cout << longestString(x, 3, 2) << endl;   // prints:  question
   return 0;
}
```

Answer:

```
string longestString(string x[][2], int rows, int cols) {
   string ans = "";
   for (int i = 0; i < rows; i++)
      for (int j = 0; j < cols; j++)
         if (x[i][j].length() > ans.length()) ans = x[i][j];
   return ans;
}
```

Problem 5 Write a function called *become5* that has two inputs – the first input is a positive integer and the second input is a single-digit integer. (You may assume that the two inputs have these forms.) The function has an integer output. The output is identical to the first input, except that every digit that matches the second input is replaced with a 5.

For example, a program that uses the function *become5* follows.

```
int main() {
   cout << become5(232, 2) << endl;       // prints 535
   cout << become5(232, 3) << endl;       // prints 252
   cout << become5(232, 4) << endl;       // prints 232
   return 0;
}
```

Answer:

```
int become5(int n, int digit) {
   if (n == digit) return 5;
   if (n < 10) return n;
   return 10 * become5(n/10, digit) + become5(n % 10, digit);
}
```

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 opens an input file called input14a.txt that contains only integers, including at least one negative integer. (You may assume that the file has exactly this content.)

2. It reads integers from the file until a negative integer is found.

3. It reports how many integers were read (upto and including the first negative value).

For example if the file input14a.txt has the following content:

```
12 16 29
17 10001
2 -34
-1 35 -3
11
```

The first negative entry in the file is its 7^{th} number -34 and the program would output: 7

Answer:

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

int main() {
  ifstream f;
  f.open("input14a.txt");
  int x = 0, count = 0;

  while (x >= 0) {
     f >> x;
     count++;
  }
  cout << count << endl;
  f.close();
  return 0;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version B Exam Fall 2014 12.23.14
Solutions
11.00am – 01.00pm, Tuesday, December 23, 2014

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 a[3] = {1.0, 1.0, 1.0}, i = 7.0, j = 8.0, k = 9.9;
   double b[5] = {1.9, 9.9, 6.9, 8.9, 3.9};
   double x[2][2] = {{2.9, 0.9}, {4.9, 8.9}};
   cout << max(i, j, k) << endl;   // (a) prints: 9.9
   printMax(b, 5);                 // (b) prints: 9.9
   cout << max2d(x, 2, 2) << endl; // (c) prints: 8.9
   swap (i, j);                    // (d) swaps i and j
   swapArrays (a, b, 2); // (e) swaps first 2 elements of arrays a and b
   return 0;
}
```

(a) Title line for **max**.
Answer:

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

(b) Title line for **printMax**.
Answer:

```
void printMax(double x[], int capacity)
```

(c) Title line for **max2d**.
Answer:

```
double max2d(double x[][2], int r, int c)
```

(d) Title line for **swap**.
Answer:

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

(e) Title line for **swapArray**.
Answer:

```
void swapArrays(double x[], double y[], int number)
```

Problem 2 Consider the following C++ program.

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

void yesNo(bool ans) {
   if (ans) cout << "Y";
   else cout << "N";
   cout << endl;
}

int main() {
    int x = 3, y = 5, z = 4, a[4] = {3, 2, 1, 0};
    if (x == y) cout << "Y\n";                              // line (a)
    if (x == a[0]) cout << "Y\n";                           // line (b)
    if (!(y < x)) cout << "Y\n"; else cout << "N\n";        // line (c)
    yesNo((x < z) && (y < z));                              // line (d)
    yesNo((x < z) || (y < z));                              // line (e)
}
```

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

Answer:

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

Answer:

```
Y
```

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

Answer:

```
Y
```

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

Answer:

```
N
```

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

Answer:

```
Y
```

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 x = 5;
      double e = 2.718;
      double a[4] = {1.0, 2.0, -3.0, -4.0};
      double b[2] = {5.5, 4.5};
// (a) Changes the sign. Here to -5
      changeSign(x);
// (b) Return first digit after decimal point. Here 7 is printed.
      cout << firstDecimal(e) << endl;
// (c) Return the number of negative entries.  Here 2 is printed.
      cout << numberNeg(a, 4) << endl;
// (d) Test whether the first argument is a factor of the second.  Here: Yes
      if (isFactor(7, 14)) cout << "Yes\n";
// (e) print average of all entries both arrays:  Here 1.0 is printed.
      averageArrays(a, 4, b, 2);
      return 0;
}
```

Answer:

(a)

```
void changeSign(int &x) {
    x =  -x;
}
```

(b)

```
int firstDecimal(double x) {
    int tenX = (int) (x * 10);
    return tenX % 10;
}
```

(c)

```
int numberNeg(double x[], int capacity) {
    int ans = 0;
    for (int i = 0; i < capacity; i++)
        if (x[i] < 0) ans++;
    return ans;
}
```

(d)

```
bool isFactor(int x, int y) {
   return y % x == 0;
}
```

(e)

```
void averageArrays(double x[], int capacityX, double y[], int capacityY) {
   double sum = 0.0;
   for (int i = 0; i < capacityX; i++) sum += x[i];
   for (int i = 0; i < capacityY; i++) sum += y[i];
   cout << sum / (capacityX + capacityY) << endl;
}
```

Problem 4 Write a function called *print3* that prints the elements of an array of integers, separated by commas and with 3 elements on each output line.

For example, a program that uses the function *print3* follows.

```
int main() {
   int x[8] = {1,2,3,4,5,6,7,8};
   print3(x, 8);
   return 0;
}
```

The output should be exactly:

```
1,2,3
4,5,6
7,8
```

Answer:

```
void print3(int x[], int capacity) {
   for (int i = 0; i < capacity; i++) {
      cout << x[i];
      if (i < (capacity - 1) && i % 3 != 2) cout << ",";
      else cout << endl;
   }
}
```

Problem 5 Write a function called *change5* that has two inputs – the first input is a positive integer and the second input is a single-digit integer. (You may assume that the two inputs have these forms.) The function has an integer output. The output is identical to the first input, except that every digit equal to 5 is replaced by the digit given by the second parameter.

For example, a program that uses the function *change5* follows.

```
int main() {
   cout << change5(535, 2) << endl;       // prints 232
   cout << change5(252, 3) << endl;       // prints 232
   cout << change5(232, 4) << endl;       // prints 232
   return 0;
}
```

Answer:

```
int change5(int n, int digit) {
   if (n == 5) return digit;
   if (n < 10) return n;
   return 10 * change5(n/10, digit) + change5(n % 10, digit);
}
```

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 opens an input file called input14b.txt that contains only strings, including at least one that starts with the letter X. (You may assume that the file has exactly this content.)

2. It reads strings from the file until one beginning with X is found.

3. It reports how many strings were read (upto and including the first that begins with X).

For example if the file input14b.txt has the following content:

```
A BBB Cat

Dog
XYZ E   XXX
```

The first X-word in the file is its 5^{th} string XYZ and the program would output: 5

Answer:

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

int main() {
  ifstream f;
  f.open("input14b.txt");
  int count = 0;
  string x = "A";

  while (x[0] != 'X') {
     f >> x;
     count++;
  }
  cout << count << endl;
  f.close();
  return 0;
}
```