QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam Spring 2016 05.23.16

Solutions

08.30am – 10.30am, Monday, May 23, 2016

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 = 0.0, y = 3.1, z = 2.5;
   int array[5] = {3,1,4,1,5};
   string s;

   cout << middle(x, y, z) << endl;              // (a) prints middle value 2.5
   increase(x); cout << x << endl;                     // (b) prints 1.0
   printBoth(y, z);                                    // (c) prints 3.1 2.5
   s = allOf(array, 5); cout << s << endl;             // (d) prints 3 1 4 1 5
   increase(array, 5); cout << allOf(array,5) << endl; // (e) prints 4 2 5 2 6
   return 0;
}
```

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

Answer:

```
double middle(double a, double b, double c)
```

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

Answer:

```
void increase(double &x)
```

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

Answer:

```
void printBoth(double a, double b)
```

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

Answer:

```
string allOf(int a[], int cap)
```

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

Answer:

```
void increase(int x[], int cap)
```

Problem 2 Consider the following C++ program.

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

string fun(string x) {
  if (x.length() <= 4) {
     return "00";
  }
  return fun(x.substr(4)) + x.substr(4);
}

int main() {
    int x = 43;
    int y = x / 10;
    cout << x / 10 + x % 10 << endl;           // line (a)
    if (((x > 40) || (x < 50)) && ((y > 4) || (y < 5)))
       cout << x % y << endl;                  // line (b)
    cout << fun("Easy") << endl;               // line (c)
    cout << fun("12345") << endl;              // line (d)
    cout << fun("123456789") << endl;            // line (e)
}
```

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

Answer:

7

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

Answer:

3

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

Answer:

00

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

Answer:

005

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

Answer:

00956789

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[4] = {1, 2 ,-3, -4};
      int x = 5, y = 6;
// (a) Return the cube. Here 8 is printed.
      cout << cube(2) << endl;
// (b) Return the larger number. Here 6 is printed.
      cout << larger(x, y) << endl;
// (c) Return the largest element. Here 2 is printed.
      cout << largest(a, 4) << endl;
// (d) Test whether all array entries are positive.  Here: Not all positive
      if (!allPositive(a, 4)) cout << "Not all positive\n";
// (e) Swap values.  Here -3 is printed.
      swap(a[2], x);
      cout << x << endl;
      return 0;
}
```

Answer:

(a)

```
int cube(int x) {
    return x * x * x;
}
```

(b)

```
int larger(int x, int y) {
    if (x > y) return x;
    return y;
}
```

(c)

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

(d)

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

(e)

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

Problem 4 Write a function called *evenCols* that returns the number of columns of a 2-dimensional array that have an even sum. The array contains integers and has 5 columns.

For example, a program that uses the function *evenCols* follows. The output is 2 because only columns 1 and 4 have even sum.

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

Answer:

```
int evenCols(int array[][5], int rows, int cols) {
   int ans = 0;
   for (int c = 0; c < cols; c++) {
      int total = 0;
      for (int r = 0; r < rows; r++)
         total += array[r][c];
      if (total % 2  == 0) ans++;
   }
   return ans;
}
```

Problem 5 Write a function called *not7s* that counts how many digits are not equal to 7 in a positive integer parameter.

For example, a program that uses the function *not7s* follows.

```
int main() {
   cout << not7s(747) << endl;          // prints 1
   cout << not7s(176) << endl;          // prints 2
   cout << not7s(12345) << endl;        // prints 5
   cout << not7s(77777) << endl;        // prints 0
   return 0;
}
```

Answer:

```
int not7s(int x) {
   if (x == 0) return 0;
   if (x % 10 == 7) return not7s(x/10);
   return not7s(x/10) + 1;
}
```

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 an integer n that is between 1 and 23.

2. It repeatedly reads n from the user until the supplied value of n is legal.

3. It prints out a rectangular picture with $2n - 1$ rows and n columns that makes a large 5 as displayed by a digital clock.

Here is an example of how the program should work:

```
Give me an integer between 1 and 23:   4
****
*
*
****
   *
   *
****
```

Answer:

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

int main() {
   int n;
   cout << "Give me an integer between 1 and 23:  ";
   cin >> n;
   while (n < 1 || n > 23) {
      cout << "Give me an integer between 1 and 23:  ";
      cin >> n;
   }

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

   return 0;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam Spring 2016 05.23.16

Solutions

08.30am – 10.30am, Monday, May 23, 2016

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 = 3, z = 2;
   char array[5] = {'a','b','c','d','e'};
   string s;

   cout << biggest(x, y, z) << endl;                     // (a) prints biggest: 3
   x = increase(x); cout << x << endl;                   // (b) prints 1
   s = printBoth(y, z); cout << s << endl;               // (c) prints 3 2
   allOf(array, 5);                                      // (d) prints a b c d e
   upper(array, 5); allOf(array,5);                      // (e) prints A B C D E
   return 0;
}
```

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

Answer:

```
int biggest(int a, int b, int c)
```

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

Answer:

```
int increase(int x)
```

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

Answer:

```
string printBoth(int a, int b)
```

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

Answer:

```
void allOf(char a[], int cap)
```

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

Answer:

```
void upper(char x[], int cap)
```

Problem 2 Consider the following C++ program.

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

string fun(string x) {
  if (x.length() <= 4) {
     return "XX";
  }
  return fun(x.substr(3)) + x.substr(4);
}

int main() {
    int x = 34;
    int y = x / 10;
    cout << x / 10 + x % 10 << endl;           // line (a)
    if (((x > 30) && (x < 50)) || ((y > 3) && (y < 5)))
       cout << x % y << endl;                  // line (b)
    cout << fun("Easy") << endl;               // line (c)
    cout << fun("ABCDE") << endl;              // line (d)
    cout << fun("ABCDEFG") << endl;            // line (e)
}
```

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

Answer:

```
7
```

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

Answer:

```
1
```

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

Answer:

```
XX
```

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

Answer:

```
XXE
```

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

Answer:

```
XXEFG
```

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[4] = {1, 2 ,-3, -4};
      int x = 5, y = 6;
// (a) Return the cube. Here 8.0 is printed.
      cout << cube(2.0) << endl;
// (b) Print the larger number. Here 6 is printed.
      larger(x, y);
// (c) Return the first negative element, or 0 if there is none. Here -3 is printed.
      cout << firstNegative(a, 4) << endl;
// (d) Test whether array entries increase in size.  Here: Not increasing
      if (!increasing(a, 4)) cout << "Not increasing\n";
// (e) Swap values.  Here 6 is printed.
      swap(y, x);
      cout << x << endl;
      return 0;
}
```

Answer:

(a)

```
double cube(double x) {
    return x * x * x;
}
```

(b)

```
void larger(int x, int y) {
    if (x > y) cout << x << endl;
    else cout << y << endl;
}
```

(c)

```
int firstNegative(int x[], int cap) {
    for (int i = 0; i < cap; i++)
       if (x[i] < 0) return x[i];
    return 0;
}
```

(d)

```
bool increasing(int x[], int capacity) {
   for (int i = 1; i < capacity; i++)
      if (x[i - 1] >= x[i]) return false;
   return true;
}
```

(e)

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

Problem 4 Write a function called *positiveCols* that returns the number of columns of a 2-dimensional array that have a positive sum. The array contains doubles and has 6 columns.

For example, a program that uses the function *positiveCols* follows. The output is 2 because only columns 1 and 3 have positive sum.

```
int main() {
   double x[2][6] = {{1.0, 6.0, 3.0, 5.0, 4.0, 2.0},
                     {-4.0, -4.0, -4.0, -4.0, -4.0, -4.0}};
   cout << positiveCols(x, 2, 6) << endl;       // prints 2
   return 0;
}
```

Answer:

```
int positiveCols(double array[][6], int rows, int cols) {
   int ans = 0;
   for (int c = 0; c < cols; c++) {
      double total = 0;
      for (int r = 0; r < rows; r++)
         total += array[r][c];
      if (total > 0) ans++;
   }
   return ans;
}
```

Problem 5 Write a function called *sixesAndSevens* that counts how many digits are equal to 6 or 7 in a positive integer parameter.

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

```
int main() {
   cout << sixesAndSevens(747) << endl;          // prints 2
   cout << sixesAndSevens(176) << endl;          // prints 2
   cout << sixesAndSevens(666) << endl;          // prints 3
   cout << sixesAndSevens(12345) << endl;        // prints 0
   return 0;
}
```

Answer:

```
int sixesAndSevens(int x) {
   if (x == 0) return 0;
   if (x % 10 == 7 || x% 10 == 6) return sixesAndSevens(x/10) + 1;
   return sixesAndSevens(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 an integer n that is between 1 and 17.

2. It repeatedly reads n from the user until the supplied value of n is legal.

3. It prints out a rectangular picture with $2n - 1$ rows and n columns that makes a large 2 as displayed by a digital clock.

Here is an example of how the program should work:

```
Give me an integer between 1 and 17:   5
*****
    *
    *
    *
*****
*
*
*
*****
```

Answer:

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

int main() {
   int n;
   cout << "Give me an integer between 1 and 17:  ";
   cin >> n;
   while (n < 1 || n > 17) {
      cout << "Give me an integer between 1 and 17:  ";
      cin >> n;
   }

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

   return 0;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam Spring 2016 05.23.16

Solutions

01.45pm – 03.45pm, Monday, May 23, 2016

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 = 3, z = 2;
   string array[5] = {"A","B","C","D","E"};
   string s;

   cout << least(x, y, z) << endl;                      // (a) prints least: 0
   x = decrease(y); cout << x << " " << y << endl;      // (b) prints 2 2
   s = printBoth(z, z); cout << s << endl;              // (c) prints 2 2
   allOf(array, 5);                                     // (d) prints A B C D E
   lower(array, 5); allOf(array,5);                     // (e) prints a b c d e
   return 0;
}
```

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

```
int least(int a, int b, int c)
```

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

```
int decrease(int &x)
```

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

```
string printBoth(int a, int b)
```

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

```
void allOf(string a[], int cap)
```

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

```
void lower(string x[], int cap)
```

Problem 2 Consider the following C++ program.

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

string fun(string x) {
  if (x.length() <= 5) {
     return "00";
  }
  return fun(x.substr(5, 1)) + x.substr(5, 1);
}

int main() {
    int x = 78;
    string y = "Hello";
    cout << x / 10 + x % 10 << endl;           // line (a)
    cout << y.find("l") << endl;               // line (b)
    cout << fun("Easy") << endl;               // line (c)
    cout << fun("234567") << endl;             // line (d)
    cout << fun("23456789") << endl;           // line (e)
}
```

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

```
15
```

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

```
2
```

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

```
00
```

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

```
007
```

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

```
007
```

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[4] = {3, 2 ,-3, -4};
      double x = 5.0, y = 6.0;
// (a) Return the cube. Here 8.0 is printed.
      cout << cube(2.0) << endl;
// (b) Print the larger number. Here 6.0 is printed.
      larger(x, y);
// (c) Return the last positive element, or 0 if there is none. Here 2 is printed.
      cout << lastPositive(a, 4) << endl;
// (d) Test whether array entries decrease in size.  Here: decreasing
      if (decreasing(a, 4)) cout << "Decreasing\n";
// (e) Swap values.  Here 2 is printed.
      swap(a[0], a[1]);
      cout << a[0] << endl;
      return 0;
}
```

Answer:

(a)

```
double cube(double x) {
    return x * x * x;
}
```

(b)

```
void larger(double x, double y) {
    if (x > y) cout << x << endl;
    else cout << y << endl;
}
```

(c)

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

(d)

```
bool decreasing(int x[], int capacity) {
   for (int i = 1; i < capacity; i++)
      if (x[i - 1] <= x[i]) return false;
   return true;
}
```

(e)

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

Problem 4 Write a function called *largestCol* that returns the largest sum of the entries in a single column of a 2-dimensional array. The array contains integers and has 5 columns.

For example, a program that uses the function *largestCol* follows. The output is 7 because this is the sum for columns 0 and 4 and the other columns have a smaller sum.

```
int main() {
   int x[2][5] = {{1, 2, 3, 5, 4}, {6, 0, 0, 0, 3}};
   cout << largestCol(x, 2, 5) << endl;       // prints 7
   return 0;
}
```

Answer:

```
int largestCol(int array[][5], int rows, int cols) {
   int ans;
   for (int c = 0; c < cols; c++) {
      int total = 0;
      for (int r = 0; r < rows; r++)
         total += array[r][c];
      if (c == 0 || total > ans) ans = total;
   }
   return ans;
}
```

Problem 5 Write a function called *diff2* that returns the absolute value of the difference of the first two digits in an integer parameter that is at least 10.

For example, a program that uses the function *diff2* follows.

```
int main() {
   cout << diff2(747) << endl;          // prints 3
   cout << diff2(176) << endl;          // prints 6
   cout << diff2(10101) << endl;        // prints 1
   cout << diff2(77777) << endl;        // prints 0
   return 0;
}
```

Answer:

```
int diff2(int x) {
   if (x < 100) {
      int ans = x/10 - x % 10;
      if (ans >= 0) return ans;
      return -ans;
   }
   return diff2(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 an integer n that is between 1 and 23.

2. It repeatedly reads n from the user until the supplied value of n is legal.

3. It prints out a rectangular picture with $2n - 1$ rows and n columns that makes a large 3 as displayed by a digital clock.

Here is an example of how the program should work:

```
Give me an integer between 1 and 23:   5
*****
    *
    *
    *
*****
    *
    *
    *
*****
```

Answer:

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

int main() {
   int n;
   cout << "Give me an integer between 1 and 23:  ";
   cin >> n;
   while (n < 1 || n > 23) {
      cout << "Give me an integer between 1 and 23:  ";
      cin >> n;
   }

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

   return 0;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam Spring 2016 05.23.16

Solutions

01.45pm – 03.45pm, Monday, May 23, 2016

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 = 0.0, y = 3.1, z = 2.5;
   int array[5] = {3,1,4,1,5};
   string s;

   cout << second(x, x, z) << endl;              // (a) prints second value 0.0
   increase(x); cout << x << endl;                     // (b) prints 1.0
   printBoth(y, z);                                    // (c) prints 3.1 2.5
   s = allOf(array, 5); cout << s << endl;             // (d) prints 3 1 4 1 5
   rotate(array, 5); cout << allOf(array,5) << endl;   // (e) prints 1 4 1 5 3
   return 0;
}
```

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

Answer:

```
double second(double a, double b, double c)
```

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

Answer:

```
void increase(double &x)
```

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

Answer:

```
void printBoth(double a, double b)
```

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

Answer:

```
string allOf(int a[], int cap)
```

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

Answer:

```
void rotate(int x[], int cap)
```

Problem 2 Consider the following C++ program.

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

string fun(string x) {
  if (x.length() <= 3) {
     return "XX";
  }
  return fun(x.substr(1,2)) + x.substr(1,2);
}

int main() {
    int x = 53;
    string y = "easy";
    cout << x / 10 + x % 10 << endl;            // line (a)
    cout << y.rfind("a") << endl;               // line (b)
    cout << fun(y) << endl;                     // line (c)
    cout << fun("y") << endl;                   // line (d)
    cout << fun("yxwvuts") << endl;             // line (e)
}
```

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

Answer:

```
8
```

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

Answer:

```
1
```

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

Answer:

```
XXas
```

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

Answer:

```
XX
```

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

Answer:

```
XXxw
```

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[4] = {3, 2 ,-3, -4};
      int x = 7, y = 6;
// (a) Return the cube. Here 8 is printed.
      cout << cube(2) << endl;
// (b) Is x larger than y?. Here YES is printed.
      if (larger(x, y)) cout << "YES" << endl;
// (c) Return the smallest element. Here -4 is printed.
      cout << smallest(a, 4) << endl;
// (d) Test whether all array entries are negative.  Here: Not all negative
      if (!allNegative(a, 4)) cout << "Not all negative\n";
// (e) Swap values.  Here -3 is printed.
      swap(a[2], x);
      cout << x << endl;
      return 0;
}
```

Answer:

(a)

```
int cube(int x) {
    return x * x * x;
}
```

(b)

```
bool larger(int x, int y) {
   return x > y;
}
```

(c)

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

(d)

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

(e)

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

Problem 4 Write a function called *smallestCol* that returns the smallest sum of the entries in a single column of a 2-dimensional array. The array contains doubles and has 6 columns.

For example, a program that uses the function *smallestCol* follows. The output is 7.0 because this is the sum for columns 0 and 4 and the other columns have a larger sum.

```
int main() {
   double x[2][6] = {{1.0, 9.0, 8.0, 6.0, 4.0, 8.0},
                     {6.0, 0.0, 0.0, 3.0, 3.0, 3.0}};
   cout << smallestCol(x, 2, 6) << endl;       // prints 7.0
   return 0;
}
```

Answer:

```
double smallestCol(double array[][6], int rows, int cols) {
   double ans;
   for (int c = 0; c < cols; c++) {
      double total = 0;
      for (int r = 0; r < rows; r++)
         total += array[r][c];
      if (c == 0 || total < ans) ans = total;
   }
   return ans;
}
```

Problem 5 Write a function called *sum3* that returns the sum of the first three digits in an integer parameter that is at least 100.

For example, a program that uses the function *sum3* follows.

```
int main() {
   cout << sum3(747) << endl;          // prints 18
   cout << sum3(176) << endl;          // prints 14
   cout << sum3(10199) << endl;        // prints  2
   cout << sum3(77777) << endl;        // prints 21
   return 0;
}
```

Answer:

```
int sum3(int x) {
   if (x == 0) return 0;
   if (x < 1000) return sum3(x/10) + x%10;
   return sum3(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 an integer n that is between 1 and 17.

2. It repeatedly reads n from the user until the supplied value of n is legal.

3. It prints out a rectangular picture with $2n - 1$ rows and n columns that makes a large 4 as displayed by a digital clock.

Here is an example of how the program should work:

```
Give me an integer between 1 and 17:   5
*   *
*   *
*   *
*****
    *
    *
    *
```

Answer:

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

int main() {
   int n;
   cout << "Give me an integer between 1 and 17:  ";
   cin >> n;
   while (n < 1 || n > 17) {
      cout << "Give me an integer between 1 and 17:  ";
      cin >> n;
   }

   for (int r = 1; r < 2*n; r++) {
      for (int c = 1; c <= n; c++)
         if (r == n) cout << "*";
         else if (r < n && c == 1) cout << "*";
         else if (c == n) cout << "*";
         else cout << " ";
      cout << endl;
   }

   return 0;
}
```