QUEENS COLLEGE Department of Computer Science
CSCI 111 Midterm 2, version A Exam Spring 2015 04.30.15

Solutions

09.00am – 09.50am, Thursday, April 30, 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() {
   string fullName = "Freddy Next Door";
   int a2[2][3] = {{-2, 4, 3}, {-3, 4, 2}};
   int a[5] = {7, 6, 5, 9, 7};
   cout << middleDigit(19683) + 1   << endl;   // (a) prints: 7 as 6 + 1
   cout << random(a2, 2, 3) << endl;           // (b) prints random entry eg 4
   cout << initials(fullName) << endl;         // (c) prints: F.N.D.
   makePositive(a2[0][0]);                     // (d) make a2[0][0] positive
   cout << number7s(a, 5);                     // (e) prints 2: the number of 7s
   return 0;
}
```

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

Answer:

```
int middleDigit(int x)
```

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

Answer:

```
int random(int [][3], int r, int c)
```

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

Answer:

```
string initials(string x)
```

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

Answer:

```
void makePositive(int &x)
```

(e) Title line for **number7s**.

Answer:

```
int number7s(int x[], int cap)
```

Problem 2 Consider the following C++ program.

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

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

int main() {
```

```
    int x = 2, y = 7, z = 10;   string s = "007";
    cout << ((double) y) / x << endl;         // line (a)
    if (!((x > y) && (y > 5))) s = "008";
    cout << s << endl;                        // line (b)
    z %= y; cout << z << endl;                // line (c)
    cout << fun(z, y) << endl;                // line (d)
    fun(x, y); cout << y - x * 2 << endl;     // line (e)
}
```

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

Answer:

```
3.5
```

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

Answer:

```
008
```

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

Answer:

```
3
```

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

Answer:

```
6
```

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

Answer:

```
1
```

Problem 3 Write a function called *removeLast0* that prints an integer parameter without its rightmost 0. If there is no 0, print the number itself. If the number is 0, print nothing.

For example, a program that uses the function *removeLast0* follows.

```
int main() {
   removeLast0(7070);        // prints 707
   cout << endl;
   removeLast0(7007);        // prints 707
   cout << endl;
   removeLast0(777);         // prints 777
   cout << endl;
   return 0;
}
```

Answer:

```
void removeLast0(int n) {
   if (n == 0) return;
   if (n% 10 == 0) cout << n/10;
   else {
       removeLast0(n/10);
       cout << n % 10;
   }
}
```

Problem 4 Write a function called *largestGap* that returns the largest gap between two adjacent elements of an array.

For example, a program that uses the function *largestGap* follows, it prints 7 since the largest gap is between the 9 and the 2.

```
int main() {
   int x[] = {3, 1, 4, 1, 5, 9, 2, 6};
   cout << largestGap(x, 8) << endl;  // prints 7
   return 0;
}
```

Answer:

```
int largestGap(int x[], int n) {
   int max = x[0] - x[1];
   for (int i = 1; i < n; i++) {
      if (x[i] - x[i - 1] > max) max = x[i] - x[i - 1];
      if (x[i - 1] - x[i] > max) max = x[i - 1] - x[i];
   }
   return max;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Midterm 1, version B Exam Spring 2015 04.30.15

Solutions

11.10am – 12.00noon, Thursday, April 30, 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() {
   string fullName = "Freddy Next Door";
   int a2[2][3] = {{-2, 4, 3}, {-3, 4, 2}};
   int a[5] = {7, 6, 5, 9, 7};
   cout << firstLetter(fullName) << endl;      // (a) prints: F
   cout << sumFirstCol(a2, 2, 3) << endl;      // (b) prints: -5  (as -2 + - 3).
   cout << middleName(fullName) << endl;       // (c) prints: Next
   makeRandom(a2, 2, 3);                       // (d) reset the array with random entries
   cout << round(((double) a[0])/((double) a[1])); // (e) prints 1
                                          //  the nearest integer to the ratio.
   return 0;
}
```

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

Answer:

```
char firstLetter(string x)
```

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

Answer:

```
int sumFirstCol(int [][3], int r, int c)
```

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

Answer:

```
string middleName(string x)
```

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

Answer:

```
void makeRandom(int x[][3], int r, int c)
```

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

Answer:

```
int round(double x)
```

Problem 2 Consider the following C++ program.

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

int fun(int x, int &y) {
    x = x + 1;
    y = y - 1;
    return y;
}
```

```
int main() {
    int x = 3, y = 9, z = 10;   string s = "Yes";
    cout << ((double) x) / z << endl;         // line (a)
    if (!((x > y) || (y > 5))) s = "No";
    cout << s << endl;                        // line (b)
    z %= y; cout << z << endl;                // line (c)
    cout << fun(z, y) << endl;                // line (d)
    fun(x, y); cout << y - x % 2 << endl;     // line (e)
}
```

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

Answer:

```
0.3
```

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

Answer:

```
Yes
```

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

Answer:

```
1
```

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

Answer:

```
8
```

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

Answer:

```
6
```

Problem 3 Write a function called *removeLast7* that removes the rightmost 7 from an integer parameter. If there is no 7, it makes no change.

For example, a program that uses the function *removeLast7* follows.

```
int main() {
   cout << removeLast7(777)  << endl;       // prints 77
   cout << removeLast7(1727) << endl;       // prints 172
   cout << removeLast7(1234) << endl;       // prints 1234
   return 0;
}
```

Answer:

```
int removeLast7(int n) {
   if (n == 0) return 0;
   if (n % 10 == 7) return n/10;
   return 10 * removeLast7(n/10) + n%10;
}
```

Problem 4 Write a function called *smallestProduct* that returns the smallest product formed by two adjacent elements of an array.

For example, a program that uses the function *smallestProduct* follows, it prints 3 since the smallest product is between the 3 and the 1.

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

Answer:

```
int smallestProduct(int x[], int n) {
   int min = x[0] * x[1];
   for (int i = 1; i < n; i++)
      if (x[i] * x[i - 1] < min) min = x[i] * x[i - 1];
   return min;
}
```