QUEENS COLLEGE Department of Computer Science
CSCI 111 Second Midterm Exam Spring 2018 05.03.18

Solutions

05.00pm – 05.50pm, Thursday, May 3, 2018

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 = 32.1, a2[2][2] = {{3, 2}, {1, 0}};
   bool a[4];
   string name = "Freddy";
   setAs(a, 4, false);              // (a) sets array a to be all false
   cout << printTruth (a, 4);       // (b) prints: false false false false
   cout << mystery(a2, a, x, name); // (c) prints: Freddy is 32.1
   exchange(x, a2[0][0]);             // (d) exchangees the values
   goodDay(name);                   // (e) prints: Hello Freddy
   return 0;
}
```

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

Answer:

```
 void setAs (bool array[4], int cap, bool value)
```

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

Answer:

```
string printTruth (bool array[], int capacity)
```

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

Answer:

```
string mystery(double a[][2], bool b[], double c, string d)
```

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

Answer:

```
void exchange(double &a, double &b)
```

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

Answer:

```
void goodDay(string name)
```

Problem 2 Consider the following C++ program.

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

int fun(int x, int &y) {
   if (x == y) cout << y;
   if (x > y) y++;
   else x++;
   return x;
}

int main() {
    int a[6] = {3, 1, 4, 1, 5, 9};
    int b = 3, c = 4;
    cout << a[b] + a[c] << endl;                           // line (a)
    cout << fun(b, c) << endl;                             // line (b)
    for (int r = 3; r <= 5; r++) cout << fun(r, c);      // line (c)
    cout << endl;
    fun(a[5], a[4]);  cout << a[4] << endl;                // line (d)
    cout << fun(a[1], a[3]); cout << a[3] << endl;       // line (e)
}
```

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

```
6
```

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

```
4
```

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

```
4455
```

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

```
6
```

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

```
121
```

Problem 3 Write a function called *percentPositive* that returns the percentage of entries in an array that are positive. Excessively long solutions that use more than 15 lines of code may lose points.

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

```
int main() {
   int x[8] = { 1, -1, -2, -3, -4, -5, -6, -7};
   cout << percentPositive(x, 8) << " percent " << endl;   // prints 12.5 percent
          // because the 1 positive number gives 12.5%
   return 0;
}
```

Answer:

```
double percentPositive(int a[], int c) {
   int count = 0;
   for (int i = 0; i < c; i++)
      if (a[i] > 0) count++;
   return count * 100.0 / c;
}
```

Problem 4 Write a function called `lucky7`. The function has an integer parameter that is positive. It calculates an answer by turning the first 7 (from the left) in the parameter to 77.

Only one 7 gets duplicated. If there is no seven in the parameter, the answer is a copy of the parameter. If the parameter is not positive your function can return any convenient answer of your choice. Excessively long solutions that use more than 15 lines of code may lose points.

For example, a program that uses the function follows.

```
int main() {
   cout << lucky7(747) << endl;    // prints 7747
   cout << lucky7(7) << endl;      // prints 77
   cout << lucky7(1234) << endl;   // prints 1234
   cout << lucky7(172737) << endl; // prints 1772737
   return 0;
}
```

Answer:

Answer:

```
int lucky7(int x) {
   if (x <= 0) return 0;
   int ans = lucky7(x / 10);
   if ((ans == x / 10) && (x % 10 == 7)) return 100 * ans + 77;
   return 10 * ans + x % 10;
}
```