QUEENS COLLEGE Department of Computer Science
CSCI 111 Midterm 2 Exam Fall 2016 11.30.16

Solutions

09.00am – 09.50am, Wednesday, November 30, 2016

Problem 1 *(points)* 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 i = 2;
   int x[5] = {3, 1, 4, 1, 5};
   cout << max(2.1, i, 1.5) << endl;                       // (a) prints 2.1
   cout << min(x[2], x[3]) << endl;                        // (b) prints 1
   negateIt(i); cout << i + 1 << endl;                     // (c) prints -1
   printArray(x, 5);                                       // (d) prints 31415
   if (sum(sum(2.1, 6), 1) > 0) cout << "big\n";           // (e) prints big
   return 0;
}
```

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

Answer:

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

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

Answer:

```
int min(int x, int y)
```

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

Answer:

```
void negateIt(int &x)
```

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

Answer:

```
void printArray(int a[], int cap)
```

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

Answer:

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

Problem 2 *(points)* Consider the following C++ program.

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

double sum(int x[], int cap, int jump) {
  double ans = 0.0;
  for (int i = 0; i < cap; i+= jump)
     ans += x[i];
  return ans / 10.0;
}

int main() {
    int x[6] = {2, 1, 3, 0, 4, 9};
    cout << x[2] << endl;                     // line (a)
    cout << x[5/3] << endl;                   // line (b)
    cout << x[x[2]] << endl;                  // line (c)
    cout <<  sum(x, 6, 1) << endl;            // line (d)
    cout <<  sum(x, 4, 2) << endl;            // line (e)
    return 0;
}
```

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

```
3
```

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

```
1
```

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

```
0
```

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

```
1.9
```

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

```
0.5
```

Problem 3 *(points)* Write a function called *maxGap* that calculates the largest gap between adjacent entries of an array. (A gap between two numbers is the absolute value of their difference.)

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

```
int main() {
   int x[5] = {2, 9, 1, 6, 3};
   cout << maxGap(x, 5) << endl;   // prints 8 corresponding to the gap from 1 to 9.
   return 0;
}
```

Answer:

```
int maxGap(int a[], int cap) {
   int max = 0;
   for (int i = 1; i < cap; i++) {
      int gap = a[i] - a[i - 1];
      if (gap < 0) gap = -gap;
      if (gap > max) max = gap;
   }
   return max;
}
```

Problem 4 *(points)* Write a function called *secondDown* that returns the result of decreasing the second digit of a positive integer parameter by 1. (If the second digit is already 0, then the value of the parameter is returned. If the parameter is less than 10, then the function can return any answer of your choice.)

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

```
int main() {
   cout << secondDown(243)  << endl;       // prints 233
   cout << secondDown(2048) << endl;       // prints 2048
   cout << secondDown(1234) + 1 << endl;   // prints 1135
   return 0;
}
```

Answer:

```
int secondDown(int x) {
   if (x < 100) {
      if (x % 10 == 0) return x;
      else return x - 1;
   }
   return secondDown(x/10) * 10 + x % 10;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Midterm 2 Exam Fall 2016 11.30.16
Solutions
09.00am – 09.50am, Wednesday, November 30, 2016

Problem 1 *(points)* 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 i = 3;
   string s = "Hello";
   int x[5] = {2, 7, 1, 8, 2};
   cout << min(i, 2.1, x[0]) << endl;            // (a) prints: 2.1
   cout << max(x[2], 3) << endl;                 // (b) prints: 3
   cout << doubleIt(i) << endl;                  // (c) prints:  2 x 3
   hi(s);  cout << s << endl;                    // (d) prints: Hi
   cout << sum(sum(2,6,i), i, i) << endl;        // (e) prints: 17
   return 0;
}
```

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

```
double min(int x, double y, int z)
```

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

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

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

```
string doubleIt(int x)
```

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

```
void hi(string &s)
```

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

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

Problem 2 *(points)* Consider the following C++ program.

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

double sum(int x[], int cap, int jump) {
  double ans = 0.0;
  for (int i = 0; i < cap; i+= jump)
     ans += x[i];
  return ans / 5.0;
}

int main() {
    int x[6] = {5, 4, 3, 2, 1, 9};
    cout << x[3] << endl;                     // line (a)
    cout << x[5/3] << endl;                   // line (b)
    cout << x[x[3]] << endl;                  // line (c)
    cout << sum(x, 6, 1) << endl;             // line (d)
    cout << sum(x, 5, 2) << endl;             // line (e)
    return 0;
}
```

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

```
2
```

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

```
4
```

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

```
3
```

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

```
4.8
```

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

```
1.8
```

Problem 3 *(points)* Write a function called *sumGaps* that calculates the sum of the gaps between adjacent entries of an array. (A gap between two numbers is the absolute value of their difference.)

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

```
int main() {
   int x[5] = {3, 1, 4, 1, 5};
   cout << sumGaps(x, 5) << endl;   // prints 12 corresponding to the sum of gaps 2 + 3 + 3 + 4.
   return 0;
}
```

Answer:

```
int sumGaps(int a[], int cap) {
   int sum = 0;
   for (int i = 1; i < cap; i++) {
      int gap = a[i] - a[i - 1];
      if (gap < 0) gap = -gap;
      sum = sum + gap;
   }
   return sum;
}
```

Problem 4 *(points)* Write a function called *thirdDown* that returns the result of decreasing the third digit of a positive integer parameter by 1. (If the third digit is already 0, then the value of the parameter is returned. If the parameter is less than 100, then the function can return any answer of your choice.)

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

```
int main() {
   cout << thirdDown(1243)  << endl;       // prints 1233
   cout << thirdDown(12048) << endl;       // prints 12048
   cout << thirdDown(11234) + 1 << endl;   // prints 11135
   return 0;
}
```

Answer:

```
int thirdDown(int x) {
   if (x < 1000) {
      if (x % 10 == 0) return x;
      else return x - 1;
   }
   return thirdDown(x/10) * 10 + x % 10;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Midterm 2 Exam Fall 2016 11.30.16

Solutions

02.45pm – 03.35pm, Wednesday, November 30, 2016

Problem 1 *(points)* 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 i = 2;
   double x[5] = {3.0, 1.1, 4.2, 1.3, 5.4};
   cout << max(4.1, x[i] / 10, i) << endl;          // (a) prints 4.1
   cout << min(x[2], x[3]) << endl;                 // (b) prints 1.3
   squareIt(i); cout << i << endl;                  // (c) prints 4
   squareAll(x, 5); cout << x[0] << endl;           // (d) prints 9.0
   if (f(f(x[0])) > 2) cout << "+" << endl;         // (e) prints +
   return 0;
}
```

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

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

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

```
double min(double x, double y)
```

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

```
void squareIt(int &x)
```

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

```
void squareAll(double a[], int cap)
```

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

```
double f(double x)
```

Problem 2 *(points)* Consider the following C++ program.

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

void down(int x[], int cap, int gap) {
  for (int i = 0; i < cap; i+= gap)
     x[i] -= gap;
}

int main() {
    int x[6] = {3, 1, 4, 1, 5, 9};
    cout << x[5] / 4 << endl;                 // line (a)
    cout << x[5/4] << endl;                   // line (b)
    cout << x[x[5]/4] << endl;                // line (c)
    down(x, 6, 1); cout << x[1] << endl;      // line (d)
    down(x, 6, 3); cout << x[1] << endl;      // line (e)
    return 0;
}
```

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

```
2
```

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

```
1
```

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

```
4
```

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

```
0
```

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

```
0
```

Problem 3 *(points)* Write a function called *evenSum* that calculates the sum of those entries in an array that are even numbers.

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

```
int main() {
   int x[8] = {3, 1, 4, 1, 5, 9, 2, 6};
   cout << evenSum(x, 8) << endl;   // prints 12
   // The even entries are 4, 2, 6 and these add to 12
   return 0;
}
```

Answer:

```
int evenSum(int a[], int cap) {
   int sum = 0;
   for (int i = 1; i < cap; i++) {
      if (a[i] % 2 == 0)
        sum = sum + a[i];
   }
   return sum;
}
```

Problem 4 *(points)* Write a function called *allEven* that reports whether all the digits in a positive integer parameter are even.

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

```
int main() {
   int x;
   cout << "Enter a number: ";
   cin >> x;
   if (allEven(x)) cout << "All digits are even." << endl;
   else cout << "Not all digits are even." << endl;
   return 0;
}
```

If the user entered any of 2, 242 or 2048, the program would print *All digits are even.* But if the user entered any of 1, 21, 1248 or 555, the program would print *Not all digits are even.*

Answer:

```
bool allEven(int x) {
   if (x % 2 == 1) return false;
   if (x < 10) return true;
   return allEven(x/10);
}
```