QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version A Exam Spring 2014 05.20.14

Solutions

08.30am – 10.30am, Tuesday, May 20, 2014

Problem 1 Write **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, i) << endl;                // (a) prints 2.1
   cout << min(x[2], x[3]) << endl;               // (b) prints 1
   doubleIt(i); cout << i << endl;                // (c) prints 4
   printIt(x, 3);                                 // (d) prints 314
   cout << sum(sum(2,6), sum(x[0],x[1])) << endl; // (e) prints 12
   return 0;
}
```

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

Answer:

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

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

Answer:

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

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

Answer:

```
void doubleIt(int &x)
```

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

Answer:

```
void printIt(int x[], int n)
```

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

Answer:

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

Problem 2 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out 123**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"An ", "easy ", "question ", ""};
    for (int i = 2; i >= 0; i--) cout << words[i];  cout << endl;      // line (a)
    for (int i = 2; i >= 0; i--) cout << words[i][i+1];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                          // line (c)
    cout << words[0][0]++ << endl;                                     // line (d)
    cout << argc << endl;                                              // line (e)
}
```

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

```
question easy An
```

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

```
ssn
```

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

```
123
```

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

```
A
```

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

```
2
```

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 = 2, b = 3, c = 4;
   ifstream f;
   string s = "HELLO";  char t[] = "HELLO";
   f.open("testFile.txt");
 // (a) Tests whether a number is seven, here No!
   if (!isSeven(c)) cout << "No!" << endl;
 // (b) Removes the last char from a string, here HELL
   cout << removeLast(s) << endl;
 // (c) Prints second word in the input file
   cout << secondWord(f) << endl;
 // (d) Print first character of a C-string, here H
   cout << firstChar(t) << endl;
 // (e) swap a with the biggest of a,b,c.  Here prints 4,3,2
   swapBig(a, b, c);
   cout  << a << b << c << endl;
   return 0;
}
```

Answer:
(a)

```
bool isSeven(int x) {
    return x == 7;
}
```

(b)

```
string removeLast(string x) {
    return x.substr(0, x.length() - 1);
}
```

(c)

```
string secondWord(ifstream &file) {
    string x;
    file >> x;
    file >> x;
    return x;
}
```

(d)

```
char firstChar(char x[]) {
   return x[0];
}
```

(e)

```
void swapBig(int &x, int &y, int &z) {
   int temp = x;
   if (x < y && z <= y) {
      x = y;
      y = temp;
   } else if (x < z) {
      x = z;
      z = temp;
   }
}
```

Problem 4 Write a function called *addMin* that calculates the minimum of the entries in an array and adds this minimum to every odd entry of the array.

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

```
int main() {
   int x[5] = {3, 1, 4, 1, 5} ;  // min is 1 here
   addMin(x, 5);
   cout << x[0] << " " << x[1] << " " << x[2] << endl;  // prints:  4 2 4
   return 0;
}
```

Answer:

```
void addMin(int x[], int cols) {
   int min  = x[0];
   for (int c = 0; c < cols; c++)
      if (x[c] < min) min = x[c];
   for (int c = 0; c < cols; c++)
      if (x[c] % 2 == 1)
         x[c] += min;
}
```

Problem 5 Write a function called *roundOff* that returns the result of turning all digits (except the first) in a positive integer parameter to 0.

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

```
int main() {
   cout << roundOff(19683) << endl;  // prints 10000
   cout << roundOff(2) << endl;      // prints 2
   return 0;
}
```

Answer:

```
int roundOff(int x) {
   if (x < 10) return x;
   return 10* roundOff(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 reads the entries in a 2-dimensional array with 5 rows and 3 columns from the user.

2. It prints the last row that has an even sum.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 3 array:
0 0 0
1 2 3
1 1 1
3 3 3
1 1 1

Last row with even sum:
1 2 3
```

Answer:

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

int main() {
   int x[5][3];
   cout << "Give me the entries of a 5 x 3 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 3; j++) cin >> x[i][j];

   int sums[5] = {0, 0, 0, 0, 0};
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 3; j++) sums[i] += x[i][j];

   cout << "Last row with even sum: \n";
   for (int i = 4; i >= 0; i--) {
      if (sums[i] % 2 == 0) {
         for (int j = 0; j < 3; j++) cout << x[i][j] << " ";
         cout << endl;
         return 0;
      }
   }
   return 0;
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version B Exam Spring 2014 05.20.14

Solutions

08.30am – 10.30am, Tuesday, May 20, 2014

Problem 1 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out xyz 987**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"Not ", "very ", "difficult ", ""};
    for (int i = 0; i <= 2; i++) cout << words[i];  cout << endl;      // line (a)
    for (int i = 0; i <= 2; i++) cout << words[i][i];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                         // line (c)
    cout << ++words[0][0] << endl;                                    // line (d)
    cout << argc << endl;                                             // line (e)
}
```

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

Answer:

```
Not very difficult
```

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

Answer:

```
Nef
```

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

Answer:

```
xyz
```

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

Answer:

```
O
```

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

Answer:

```
3
```

Problem 2 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 = 123, b = 3, c = 4;
   ifstream f;
   string s = "HELLO";  char t[] = "HELLO";
   f.open("testFile.txt");
 // (a) Tests whether a number has 3 digits, here Yes!
   if (is3digit(a)) cout << "Yes!" << endl;
 // (b) Returns the part of a string before its midpoint, here HE
```

```
   cout << halfIt(s) << endl;
 // (c) The number of characters read from the input file before eof() is true
   cout << countChar(f) << endl;
 // (d) Print third character of a C-string that has a middle, here L
   cout << thirdChar(t) << endl;
 // (e) Replace a, b and c by their sum to print 130, 130, 130
   replace(a, b, c);
   cout  << a << "," << b << "," << c << endl;
   return 0;
}
```

Answer:

(a)

```
bool is3digit(int x) {
    return (x > 99) && (x < 1000);
}
```

(b)

```
string halfIt(string x) {
    return x.substr(0, x.length()/2);
}
```

(c)

```
int countChar(ifstream &file) {
    char x;
    int count = 0;
    while (!file.eof()) {
        x = file.get();
        count++;
    }
    return count;
}
```

(d)

```
char thirdChar(char x[]) {
   return x[2];
}
```

(e)

```
void replace(int &x, int &y, int &z) {
   x = x + y + z;
   y = x;
   z = x;
}
```

Problem 3 Write a function called *subtractAverage* that calculates the average of the entries in a 2-dimensional array (that is known to have 2 columns) and subtracts this average from every entry of the array.

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

```
int main() {
   double x[3][2] = {{1,3}, {1,3}, {1,3}} ;  // average is 2 here
   subtractAverage(x, 3, 2);
   cout << x[0][0] << " " << x[0][1] << endl;  // prints:  -1 1
   return 0;
}
```

Answer:

```
void subtractAverage(double x[][2], int rows, int cols) {
   double sum  = 0;
   for (int r = 0; r < rows; r++)
      for (int c = 0; c < cols; c++) sum += x[r][c];
   double average = sum / (rows * cols);
   for (int r = 0; r < rows; r++)
      for (int c = 0; c < cols; c++) x[r][c] -= average;

}
```

Problem 4 Write a function called *allFirst* that returns the result of turning all digits in a positive integer parameter to match the first digit.

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

```
int main() {
   cout << allFirst(19683) << endl;  // prints 11111
   cout << allFirst(2048) << endl;   // prints 2222
   return 0;
}
```

Answer:

```
int allFirst(int x) {
   if (x < 10) return x;
   int y = allFirst(x/10);
   return 10*y + y%10;
}
```

Problem 5 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads the entries in a 2-dimensional array with 4 rows and 4 columns from the user.
2. It prints (all) columns that have the greatest sum.

Here is an example of how the program should work:

```
Give me the entries of a 4 x 4 array:
0 0 0 -1
1 2 3 4
1 1 1 1
2 3 3 2

Largest columns:
0 3 1 3
```

Answer:

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

int main() {
```

```
   int x[4][4];
   cout << "Give me the entries of a 4 x 4 array:" << endl;
   for (int i = 0; i < 4; i++)
      for (int j = 0; j < 4; j++) cin >> x[i][j];

   int sums[4] = {0, 0, 0 ,0 };
   for (int i = 0; i < 4; i++)
      for (int j = 0; j < 4; j++) sums[j] += x[i][j];

   int max = sums[0];
   for (int i = 1; i < 4; i++)
      if (sums[i] > max) max = sums[i];

   cout << "Largest columns\n";
   for (int j = 0; j < 4; j++) {
      if (sums[j] == max) {
         for (int i = 0; i < 4; i++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

Problem 6 Write **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;
   int x[5] = {2, 7, 1, 8, 2};
   cout << min(i, 2.1, i) << endl;                // (a) prints 2.1
   cout << max(x[2], 3) << endl;                  // (b) prints 3
   cout << doubleIt(i) << endl;                   // (c) prints the following:  2 x 3
   cout << sum(sum(2,6,i), i, i) << endl;         // (d) prints 17
   sortIt(x, 3);                                  // (e) sorts array x by selection sort
   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 **sum**.

Answer:

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

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

Answer:

```
void sortIt(int x[], int n)
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version C Exam Spring 2014 05.20.14

Solutions

08.30am – 10.30am, Tuesday, May 20, 2014

Problem 1 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 = 2, b = 3, c = 4;
   ifstream f;
   string s = "HELLO";  char t[] = "HELLO";
   f.open("testFile.txt");
 // (a) Tests whether a number is even, here Even!
   if (isEven(c)) cout << "Even!" << endl;
 // (b) Removes first and last chars from a string, here ELL
   cout << removeEnds(s) << endl;
 // (c) Prints first word in the input file
   cout << firstWord(f) << endl;
 // (d) Print last character of a C-string, here O
   cout << lastChar(t) << endl;
 // (e) Rotate a,b,c so as to print 3,4,2
   rotate(a, b, c);
   cout  << a << b << c << endl;
   return 0;
}
```

Answer:

(a)

```
bool isEven(int x) {
    return x % 2 == 0;
}
```

(b)

```
string removeEnds(string x) {
    return x.substr(1, x.length() - 2);
}
```

(c)

```
string firstWord(ifstream &file) {
    string x;
    file >> x;
    return x;
}
```

(d)

```
char lastChar(char x[]) {
   return x[strlen(x) - 1];
}
```

(e)

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

Problem 2 Write a function called *addMin* that calculates the minimum of the entries in a 2-dimensional array (that is known to have 2 columns) and adds this minimum to every entry of the array.

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

```
int main() {
   int x[3][2] = {{1,3}, {1,3}, {1,3}} ;  // min is 1 here
   addMin(x, 3, 2);
   cout << x[0][0] << " " << x[0][1] << endl;  // prints:  2 4
   return 0;
}
```

Answer:

```
void addMin(int x[][2], int rows, int cols) {
   int min  = x[0][0];
   for (int r = 0; r < rows; r++)
      for (int c = 0; c < cols; c++)
         if (x[r][c] < min) min = x[r][c];
   for (int r = 0; r < rows; r++)
      for (int c = 0; c < cols; c++)
         x[r][c] += min;
}
```

Problem 3 Write a function called *firstDown* that returns the result of decreasing the first digit in a positive integer by 1.

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

```
int main() {
   cout << firstDown(2048) << endl;       // prints 1048
   cout << firstDown(19683) << endl;      // prints 9683
   return 0;
}
```

Answer:

```
int firstDown(int x) {
   if (x < 10) return x - 1;
   return 10* firstDown(x/10) + x % 10;
}
```

Problem 4 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads the entries in a 2-dimensional array with 5 rows and 3 columns from the user.

2. It prints the last column that has an even sum.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 3 array:
0 0 0
1 2 3
1 1 1
```

```
3 3 3
1 2 0

Last column with even sum:
0 2 1 3 2
```

Answer:

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

int main() {
   int x[5][3];
   cout << "Give me the entries of a 5 x 3 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 3; j++) cin >> x[i][j];

   int sums[5] = {0, 0, 0, 0, 0};
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 3; j++) sums[j] += x[i][j];

   cout << "Last column  with even sum: \n";
   for (int i = 2; i >= 0; i--) {
      if (sums[i] % 2 == 0) {
         for (int j = 0; j < 5; j++) cout << x[j][i] << " ";
         cout << endl;
         return 0;
      }
   }
   return 0;
}
```

Problem 5 Write **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, 1, 4, 1, 5};
   cout << max(4.1, x[i], i) << endl;               // (a) prints 4.1
   cout << min(x[2], x[3]) << endl;                 // (b) prints 1
   doubleIt(i); cout << i << endl;                  // (c) prints 4
   printIt(x, 3);                                   // (d) prints 314
   cout << sum(sum(2.1,6), sum(x[0],x[1])) << endl; // (e) prints 12.1
   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 **doubleIt**.

Answer:

```
void doubleIt(int &x)
```

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

Answer:

```
void printIt(double x[], int n)
```

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

Answer:

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

Problem 6 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out 007**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"Not ", "very ", "difficult ", ""};
    for (int i = 2; i >= 0; i--) cout << words[i];  cout << endl;      // line (a)
    for (int i = 2; i >= 0; i--) cout << words[i][i+1];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                          // line (c)
    cout << words[0][0]++ << endl;                                     // line (d)
    cout << argc << endl;                                              // line (e)
}
```

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

Answer:

```
difficult very Not
```

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

Answer:

```
fro
```

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

Answer:

```
007
```

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

Answer:

```
N
```

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

Answer:

```
2
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version D Exam Spring 2014 05.20.14

Solutions

08.30am – 10.30am, Tuesday, May 20, 2014

Problem 1 Write a function called *subtractAverage* that calculates the average of the entries in an array and subtracts this average from every positive entry of the array.

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

```
int main() {
   double x[5] = {3, 1, 4, 1, 6} ;               // average is 3 here
   subtractAverage(x, 5);
   cout << x[0] << " " << x[1] << x[2] << endl;  // prints:  0 -2 1
   return 0;
}
```

Answer:

```
void subtractAverage(double x[], int capacity) {
   double sum  = 0;
   for (int r = 0; r < capacity; r++) sum += x[r];
   double average = sum / capacity;
   for (int r = 0; r < capacity; r++)
      if (x[r] > 0) x[r] -= average;

}
```

Problem 2 Write a function called *firstUp* that returns the result of increasing the first digit of the parameter by 1, unless this first digit is 9 in which case it is not changed.

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

```
int main() {
   cout << firstUp(19683) << endl;  // prints 29683
   cout << firstUp(95) << endl;     // prints 95
   return 0;
}
```

Answer:

```
int firstUp(int x) {
   if (x < 9) return x + 1;
   if (x == 9) return x;
   return 10* firstUp(x/10) + x % 10;
}
```

Problem 3 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads the entries in a 2-dimensional array with 4 rows and 4 columns from the user.

2. It prints (all) rows that have the greatest sum.

Here is an example of how the program should work:

```
Give me the entries of a 4 x 4 array:
0 0 0 -1
1 2 3 4
1 1 1 1
2 3 3 2

Largest rows:
1 2 3 4
2 3 3 2
```

Answer:

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

int main() {
   int x[4][4];
   cout << "Give me the entries of a 4 x 4 array:" << endl;
   for (int i = 0; i < 4; i++)
      for (int j = 0; j < 4; j++) cin >> x[i][j];

   int sums[4] = {0, 0, 0 ,0 };
   for (int i = 0; i < 4; i++)
      for (int j = 0; j < 4; j++) sums[i] += x[i][j];

   int max = sums[0];
   for (int i = 1; i < 4; i++)
      if (sums[i] > max) max = sums[i];

   cout << "Largest rows\n";
   for (int i = 0; i < 4; i++) {
      if (sums[i] == max) {
         for (int j = 0; j < 4; j++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

Problem 4 Write **title lines** for the functions that are called by the following main program. **Do not supply blocks for the functions.**

```
int main() {
   double i = 3;
   double x[5] = {2, 7, 1, 8, 2};
   cout << min(i, 2.1, i) << endl;                // (a) prints 2.1
   cout << max(x[2], 3.1) << endl;                // (b) prints 3.1
   cout << doubleIt(i) << endl;                   // (c) prints the following:  2 x 3
   cout << sum(sum(2.1,6,i), i, i) << endl;       // (d) prints 17.1
   sortIt(x, 3);                                  // (e) sorts array x by selection sort
   return 0;
}
```

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

Answer:

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

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

Answer:

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

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

Answer:

```
string doubleIt(double x)
```

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

Answer:

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

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

Answer:

```
void sortIt(double x[], int n)
```

Problem 5 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out abc 123**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"An ", "easy ", "question ", ""};
    for (int i = 0; i <= 2; i++) cout << words[i];  cout << endl;      // line (a)
    for (int i = 0; i <= 2; i++) cout << words[i][i];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                         // line (c)
    cout << ++words[0][0] << endl;                                    // line (d)
    cout << argc << endl;                                             // line (e)
}
```

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

Answer:

```
An easy question
```

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

Answer:

```
Aae
```

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

Answer:

```
abc
```

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

Answer:

```
B
```

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

Answer:

```
3
```

Problem 6 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 = 23, b = 3, c = 4;
   ifstream f;
   string s = "HELLO";  char t[] = "HELLO";
   f.open("testFile.txt");
 // (a) Tests whether a number has 2 digits, here Yes!
```

```
   if (is2digit(a)) cout << "Yes!" << endl;
 // (b) Doubles a string, here HELLOHELLO
   cout << doubleIt(s) << endl;
 // (c) The number of words read from the input file before eof() is true
   cout << countWords(f) << endl;
 // (d) Print middle character of a C-string that has a middle, here L
   cout << midChar(t) << endl;
 // (e) Rotate a,b,c so as to print 4,23,3
   rotate(a, b, c);
   cout  << a << "," << b << "," << c << endl;
   return 0;
}
```

Answer:

(a)

```
bool is2digit(int x) {
    return (x > 9) && (x < 100);
}
```

(b)

```
string doubleIt(string x) {
    return x + x;
}
```

(c)

```
int countWords(ifstream &file) {
    string x;
    int count = 0;
    while (!file.eof()) {
       file >> x;
       count++;
    }
    return count;
}
```

(d)

```
char midChar(char x[]) {
   return x[(strlen(x) - 1)/2];
}
```

(e)

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

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version E Exam Spring 2014 05.22.14

Solutions

11.00am – 01.00pm, Thursday, May 22, 2014

Problem 1 Write **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 << add(i, i) << endl;                // (a) prints 4
   cout << numOdd(x, 5) << endl;            // (b) prints 4
   doubleIt(x[1]); cout << x[1] << endl;     // (c) prints 2
   cout << diff(diff(3,1), 1) << endl;       // (d) prints 1
   cout << percentage(i, x[2]) << endl;      // (e) prints 50%
   return 0;
}
```

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

Answer:

```
int add(int y, int z)
```

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

Answer:

```
int numOdd(int x[], int y)
```

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

Answer:

```
void doubleIt(int &x)
```

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

Answer:

```
int diff(int x, int y)
```

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

Answer:

```
string percentage(int x, int y)
```

Problem 2 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out CS111**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"Queens ", "College ", "CUNY ", "NY"};
    for (int i = 3; i >= 0; i--) cout << words[i];  cout << endl;      // line (a)
    for (int i = 2; i >= 0; i--) cout << words[i][i+1];  cout << endl; // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                         // line (c)
    cout << words[0][0]++ << endl;                                    // line (d)
    cout << ++argc << endl;                                           // line (e)
}
```

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

Answer:

```
NYCUNY College Queens
```

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

Answer:

```
Ylu
```

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

Answer:

```
CS111
```

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

Answer:

```
Q
```

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

Answer:

```
3
```

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() {
   string s = "HELLO", t = "GOODBYE";
 // (a) Tests whether a string starts in upper case
   if (isUpper(s)) cout << "Upper Case!" << endl;
 // (b) Tests whether a string omits the letter E
   cout << hasNoE(s) << endl;
 // (c) Returns a string that drops the first character
   cout << dropFirst(t) << endl;
 // (d) Prints the last character
   cout << last(t) << endl;
 // (e) If t is shorter than s, swap the strings, otherwise do nothing
   sort(s, t);
   cout  << s << " " << t << endl;
   return 0;
}
```

Answer:

(a)

```
bool isUpper(string x) {
    'A' <= x[0] && x[0] <= 'Z';
}
```

(b)

```
bool hasNoE(string x) {
    return x.find("E") < 0;
}
```

(c)

```
string dropFirst(string x) {
    return x.substr(1);
}
```

(d)

```
char last(string x) {
   return x[x.length() - 1];
}
```

(e)

```
void sort(string &x, string &y) {
   if (x.length() <= y.length()) return;
   string temp = x;
   x = y;
   y = temp;
}
```

Problem 4 Write a function called *gapProd* that calculates the product 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 *gapProd* follows.

```
int main() {
   int x[5] = {3, 1, 4, 1, 5};
   cout << gapProd(x, 5) << endl;   // prints 72
   // The gaps are 2, 3, 3, 4 and these multiply to 72
   return 0;
}
```

Answer:

```
int gapProd(int x[], int cap) {
   int ans = 1;
   for (int i = 1; i < cap; i++) {
      if (x[i] > x[i-1]) {
            ans = ans * (x[i] - x[i - 1]);
      }  else {
            ans = ans * (x[i - 1] - x[i]);
      }
   }
   return ans;
}
```

Problem 5 Write a function called *oddOne* that returns the result of turning all odd digits in a positive integer parameter to 1.

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

```
int main() {
   cout << oddOne(19683) << endl;  // prints 11681
   cout << oddOne(2) << endl;      // prints 2
   return 0;
}
```

Answer:

```
int oddOne(int x) {
   if (x == 0) return 0;
   if (x % 2 == 0) return 10 * oddOne(x/10) + x % 10;
   return 10* oddOne(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 reads (from the user) the entries in a 2-dimensional array with 5 rows and 5 columns.

2. It prints (all) columns that have the property that entries increase as we move down their rows.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 5 array:
0 1 5  10  10
0 2 4  11  20
0 3 3   9  21
0 4 2  12  41
0 5 1  13  99

Increasing columns:
1 2 3 4 5
10 20 21 41 99
```

Answer:

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

int main() {
   int x[5][5];
   cout << "Give me the entries of a 5 x 5 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 5; j++) cin >> x[i][j];

   cout << "Increasing columns\n";
   for (int j = 0; j < 5; j++) {
      bool ok = true;
      for (int i = 1; i < 5; i++)
         if (x[i][j] <= x[i -1][j]) ok = false;
      if (ok) {
         for (int i = 0; i < 5; i++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version F Exam Spring 2014 05.22.14

Solutions

11.00am – 01.00pm, Thursday, May 22, 2014

Problem 1 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out out out**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"CS ", "QC ", "CUNY ", "EDU "};
    for (int i = 0; i <= 2; i++) cout << words[i];  cout << endl;      // line (a)
    for (int i = 0; i <= 2; i++) cout << words[i][i];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                         // line (c)
    cout << ++words[0][0] << endl;                                    // line (d)
    cout << argc++ << endl;                                           // line (e)
}
```

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

Answer:

```
CS QC CUNY
```

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

Answer:

```
CCN
```

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

Answer:

```
out
```

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

Answer:

```
D
```

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

Answer:

```
3
```

Problem 2 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() {
   string s = "HELLO", t = "GOODBYE";
 // (a) Do two strings have the same number of characters?
   cout << sameLength(s, t) << endl;
 // (b) Tests whether a string contains a target
   cout << contains("HELL", s) << endl;
 // (c) Returns a string that drops the last character
   cout << dropLast(t) << endl;
```

```
 // (d) Prints the third character
   cout << third(t) << endl;
 // (e) Turns an upper case character to lower case
   lower(s[0]);
   cout  << s << endl;
   return 0;
}
```

Answer:

(a)

```
bool sameLength(string x, string y) {
    return x.length() == y.length();
}
```

(b)

```
bool contains(string target, string x) {
    return x.find(target) >= 0;
}
```

(c)

```
string dropLast(string x) {
    return x.substr(0, x.length() - 1);
}
```

(d)

```
char third(string x) {
   return x[2];
}
```

(e)

```
void lower(char &x) {
   if ('A' <= x && x <= 'Z') x = x + 'a' - 'A';
}
```

Problem 3 Write a function called *minGap* that calculates the smallest 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 *minGap* follows.

```
int main() {
   int x[5] = {3, 1, 4, 1, 5};
   cout << minGap(x, 5) << endl;   // prints 2 corresponding to the gap from 3 to 1.
   return 0;
}
```

Answer:

```
int minGap(int x[], int cap) {
   int ans = x[1] - x[0];
   if (ans < 0) ans = -ans;
```

```
   for (int i = 1; i < cap; i++) {
      if (x[i] > x[i-1]) {
         if ((x[i] - x[i-1]) < ans)
            ans = x[i] - x[i - 1];
      }  else {
         if ((x[i- 1] - x[i]) < ans)
            ans = x[i - 1] - x[i];
      }
   }
   return ans;
}
```

Problem 4 Write a function called *oddOneOut* that returns the result of removing the rightmost odd digit in a positive integer parameter.

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

```
int main() {
   cout << oddOneOut(19682) << endl;  // prints 1682
   cout << oddOneOut(2) << endl;      // prints 2
   return 0;
}
```

Answer:

```
int oddOneOut(int x) {
   if (x == 0) return 0;
   if (x % 2 == 1) return x/10;
   return 10 * oddOneOut(x/10) + x % 10;
}
```

Problem 5 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads (from the user) the entries in a 2-dimensional array with 5 rows and 5 columns.

2. It prints (all) rows that have the property that entries decrease as we move along their columns.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 5 array:
0 0 0 0 0
1 2 3 4 5
501 5 306 107 99
2 -1 -3 -4 -5
5 4 3 2 1

Decreasing rows:
2 -1 -3 -4 -5
5 4 3 2 1
```

Answer:

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

int main() {
   int x[5][5];
   cout << "Give me the entries of a 5 x 5 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 5; j++) cin >> x[i][j];
```

```
   cout << "Decreasing rows\n";
   for (int i = 0; i < 5; i++) {
      bool ok = true;
      for (int j = 1; j < 5; j++)
         if (x[i][j] >= x[i][j - 1]) ok = false;
      if (ok) {
         for (int j = 0; j < 5; j++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

Problem 6 Write **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 << average(x, 5) << endl;             // (a) prints 2.8
   cout << max(i, i, 3) << endl;              // (b) prints 3
   cout << doubleIt(x[1]) << endl;            // (c) prints 2
   cout << total(total(3,1,1), 1, 1) << endl; // (d) prints 7
   percentage(i, x[2]);                       // (e) prints 50%
   return 0;
}
```

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

Answer:

```
double average(int y[], int cap)
```

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

Answer:

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

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

Answer:

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

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

Answer:

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

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

Answer:

```
void percentage(int x, int y)
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final Exam, version G Exam Spring 2014 05.22.14

Solutions

11.00am – 01.00pm, Thursday, May 22, 2014

Problem 1 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() {
   string s = "HELLO", t = "GOODBYE";
 // (a) Tests whether a string has 5 or more letters
   if (isLong(s)) cout << "Long!" << endl;
 // (b) Tests whether a string contains the letter E
   cout << hasE(s) << endl;
 // (c) Returns a string with just the first 4 characters
   cout << first4(t) << endl;
 // (d) Prints the last character at or before the middle of the string
   cout << middle(t) << endl;
 // (e) swaps them
   swap(s, t);
   cout  << s << " " << t << endl;
   return 0;
}
```

Answer:

(a)

```
bool isLong(string x) {
    return x.length() > 4;
}
```

(b)

```
bool hasE(string x) {
    return x.find("E") >= 0;
}
```

(c)

```
string first4(string x) {
    return x.substr(0,4);
}
```

(d)

```
char middle(string x) {
   return x[x.length()/2];
}
```

(e)

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

Problem 2 Write a function called *gapSum* 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 *gapSum* follows.

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

Answer:

```
int gapSum(int x[], int cap) {
   int ans = 0;
   for (int i = 1; i < cap; i++) {
      if (x[i] > x[i-1]) {
            ans = ans + x[i] - x[i - 1];
      }  else {
            ans = ans + x[i - 1] - x[i];
      }
   }
   return ans;
}
```

Problem 3 Write a function called *eveNine* that returns the result of turning all even digits in a positive integer parameter to 9.

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

```
int main() {
   cout << eveNine(19683) << endl;  // prints 19993
   cout << eveNine(3) << endl;      // prints 3
   return 0;
}
```

Answer:

```
int eveNine(int x) {
   if (x == 0) return 0;
   if (x % 2 != 0) return 10 * eveNine(x/10) + x % 10;
   return 10* eveNine(x/10) + 9;
}
```

Problem 4 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads (from the user) the entries in a 2-dimensional array with 5 rows and 5 columns.

2. It prints (all) columns that have the property that entries decrease as we move down their rows.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 5 array:
0 1 5  10  99
0 2 4  11  41
0 3 3   9  21
0 4 2  12  20
0 5 1  13  10

Decreasing columns:
5 4 3 2 1
99 41 21 20 10
```

Answer:

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

int main() {
   int x[5][5];
   cout << "Give me the entries of a 5 x 5 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 5; j++) cin >> x[i][j];

   cout << "Decreasing columns\n";
   for (int j = 0; j < 5; j++) {
      bool ok = true;
      for (int i = 1; i < 5; i++)
         if (x[i][j] >= x[i -1][j]) ok = false;
      if (ok) {
         for (int i = 0; i < 5; i++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

Problem 5 Write **title lines** for the functions that are called by the following main program. **Do not supply blocks for the functions.**

```
int main() {
   double i = 2.5;
   int x[5] = {3, 1, 4, 1, 5};
   cout << add(i, i) << endl;                // (a) prints 5.0
   if (oddSum(x, 5)) cout << "true" << endl; // (b) prints true
   doubleIt(i); cout << i << endl;           // (c) prints 5.0
   cout << diff(diff(3.0,i), i) << endl;     // (d) prints -2.0
   cout << percentage(x[1], x[2]) << endl;   // (e) prints 25%
   return 0;
}
```

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

Answer:

```
double add(double y, double z)
```

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

Answer:

```
bool oddSum(int x[], int y)
```

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

Answer:

```
void doubleIt(double &x)
```

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

Answer:

```
double diff(double x, double y)
```

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

Answer:

```
string percentage(int x, int y)
```

Problem 6 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out 007**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"Queens ", "College ", "Flushing ", "New York"};
    for (int i = 3; i >= 0; i--) cout << words[i];  cout << endl;      // line (a)
    for (int i = 3; i >= 0; i--) cout << words[i][i+1];  cout << endl;   // line (b)
    words[3] = argv[1];
    cout << words[3] << endl;                                          // line (c)
    cout << words[0][0]++ << endl;                                     // line (d)
    cout << --argc << endl;                                              // line (e)
}
```

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

Answer:

```
New YorkFlushing College Queens
```

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

Answer:

```
Yslu
```

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

Answer:

```
007
```

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

Answer:

```
Q
```

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

Answer:

```
1
```

QUEENS COLLEGE Department of Computer Science
CSCI 111 Final, version H Exam Spring 2014 05.22.14

Solutions

11.00am – 01.00pm, Thursday, May 22, 2014

Problem 1 Write a function called *maxGap* that calculates the biggest 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] = {3, 1, 4, 1, 5};
   cout << maxGap(x, 5) << endl;   // prints 4 corresponding to the gap from 1 to 5.
   return 0;
}
```

Answer:

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

Problem 2 Write a function called *evenOut* that returns the result of removing the rightmost even digit in a positive integer parameter.

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

```
int main() {
   cout << evenOut(19683) << endl;  // prints 1963
   cout << evenOut(2) << endl;      // prints 0
   return 0;
}
```

Answer:

```
int evenOut(int x) {
   if (x == 0) return 0;
   if (x % 2 == 0) return x/10;
   return 10 * evenOut(x/10) + x % 10;
}
```

Problem 3 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It reads (from the user) the entries in a 2-dimensional array with 5 rows and 5 columns.

2. It prints (all) rows that have the property that entries increase as we move along their columns.

Here is an example of how the program should work:

```
Give me the entries of a 5 x 5 array:
0 0 0 0 0
1 2 3 4 5
1 5 6 7 99
2 -1 3 4 5
```

```
5 4 3 2 1

Increasing rows:
1 2 3 4 5
1 5 6 7 99
```

Answer:

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

int main() {
   int x[5][5];
   cout << "Give me the entries of a 5 x 5 array:" << endl;
   for (int i = 0; i < 5; i++)
      for (int j = 0; j < 5; j++) cin >> x[i][j];

   cout << "Increasing rows\n";
   for (int i = 0; i < 5; i++) {
      bool ok = true;
      for (int j = 1; j < 5; j++)
         if (x[i][j] <= x[i][j - 1]) ok = false;
      if (ok) {
         for (int j = 0; j < 5; j++) cout << x[i][j] << " ";
         cout << endl;
      }
   }
}
```

Problem 4 Write **title lines** for the functions that are called by the following main program. **Do not supply blocks for the functions.**

```
int main() {
   double i = 2; int n = 2;
   double x[5] = {3, 1, 4, 1, 5};
   cout << average(x, 5) << endl;             // (a) prints 2.8
   cout << max(i, i, 3.0) << endl;            // (b) prints 3.0
   cout << doubleIt(x[1]) << endl;            // (c) prints 2.0
   cout << ratio(ratio(3,1), n) << endl;      // (d) prints 1.5
   percentage(i, x[2]);                       // (e) prints 50.0%
   return 0;
}
```

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

Answer:

```
double average(double y[], int cap)
```

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

Answer:

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

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

Answer:

```
double doubleIt(double x)
```

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

Answer:

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

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

Answer:

```
void percentage(double x, double y)
```

Problem 5 Consider the following C++ program. It is compiled to **a.out** and executed with the command **./a.out a 1**.

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

int main(int argc, char *argv[]) {
    string words[4] = {"CS111 ", "Queens ", "College ", ""};
    for (int i = 1; i <= 3; i++) cout << words[i];  cout << endl;      // line (a)
    for (int i = 0; i <= 2; i++) cout << words[i][i];  cout << endl;   // line (b)
    words[3] = argv[2];
    cout << words[3] << endl;                                         // line (c)
    cout << ++words[0][0] << endl;                                    // line (d)
    cout << argc << endl;                                             // line (e)
}
```

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

Answer:

```
Queens College
```

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

Answer:

```
Cul
```

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

Answer:

```
1
```

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

Answer:

```
D
```

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

Answer:

```
3
```

Problem 6 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() {
   string s = "HELLO", t = "GOODBYE";
 // (a) return number of characters
   cout << stringLength(s) << endl;
 // (b) Tests whether a string contains a target
   cout << contains(s, "HELL") << endl;
```

```
 // (c) Returns a string with just the last 4 characters
   cout << last4(t) << endl;
 // (d) Prints the first character
   cout << first(t) << endl;
 // (e) adds on the second string
   addOn(s, t);
   cout  << s << endl;
   return 0;
}
```

Answer:

(a)

```
int stringLength(string x) {
    x.length();
}
```

(b)

```
bool contains(string x, string target) {
    return x.find(target) >= 0;
}
```

(c)

```
string last4(string x) {
    return x.substr(x.length() - 4, 4);
}
```

(d)

```
char first(string x) {
   return x[0];
}
```

(e)

```
void addOn(string &x, string y) {
   x = x + y;
}
```