

Solutions

09.10am – 10.00am, Wednesday, May 5, 2021

Problem 1 Write the best **title lines** for the functions that are called by the following main program. **Do not supply the blocks for the functions.**

```
int main() {
    int x = 12, y = 36, w = 21331;
    double z[4] = {1.1, -1.11, 2.5, 5.7};

    // a. The function ratio returns x/y rounded to 2 decimal places eg as 0.33.
    cout << ratio(x, y) << endl; // (a)

    // b. The function sameSign reports whether two numbers have the same sign. Here it returns false.
    if ( !sameSign(z[0], z[1]) ) cout << "Opposite signs\n"; // (b)

    // c. The function cutDuplicates removes all duplicate digits from a number.
    cutDuplicates( w ); // (c)
    cout << w << endl; // prints 2

    // d. The function inWords makes the word for a number (here twelve) from an integer parameter.
    cout << inWords(x) << endl; // (d)

    // e. A mystery function.
    mystery(mystery(sameSign(z[0], ratio(x,y)))); // (e)

    return 0;
}
```

(a) Title line for **ratio** as called at the line marked (a).

Answer: double ratio(int x, int y)

(b) Title line for **sameSign** as called at the line marked (b).

Answer: bool sameSign(double x, double y)

(c) Title line for **cutDuplicates** as called at the line marked (c).

Answer: void cutDuplicates(int &x)

(d) Title line for **inWords** as called at the line marked (d).

Answer: string inWords(int x)

(e) Title line for **mystery** as called at the line marked (e).

Answer: bool mystery(bool b)

Problem 2 Consider the following C++ program. The program makes use of a function `reverseArray` that reverses the entries in an array. So that if the array has capacity 5 and entries 1, 2, 3, 4, 5 in this order, the function `reverseArray` moves the entries so that they are ordered as 5, 4, 3, 2, 1.

Make sure to use your own 8-digit CUNY ID number as the number entered as input to the program. It would be a very bad idea to give answers based on another student's ID number!

```
int main() {
    int id, a[10] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3};

    cout << "Enter your 8-digit CUNY id number: ";
    cin >> id;    // assume that the user types YOUR OWN CUNY ID number

    cout << id << endl;           // line (a)
    cout << a[ id % 10 ] << endl;  // line (b)

    reverseArray(a, 10);
    cout << a[ a[0] ] << endl;    // line (c)
    cout << a[ 4 ] % a[ 5 ] << endl; // line (d)
    cout << a[ 0 ] + a[ 2 ] % a[ 3 ] << endl; // line (e)

    return 0;
}
```

Enter your 8-digit CUNY id number: (a) What is the output from the instruction beginning on line (a)?

Answer:

12345678

This answer is based on the ID number 12345678. Actual answers will be different.

(b) What is the output from the instruction beginning on line (b)?

Answer:

5

The answer will be the array entry indexed by the last digit of the answer to (a).

(c) What is the output from the instruction beginning on line (c)?

Answer:

2

(d) What is the output from the instruction beginning on line (d)?

Answer:

4

(e) What is the output from the instruction beginning on line (e)?

Answer:

3

Problem 3 Write a function called *sumAbsolute* that returns the sum of the absolute values of the entries in an array with base type double. (The absolute value $|x|$ of a number is obtained by ignoring its sign, so for example $|-4| = |4| = 4$.) The function should use 2 parameters as follows: the array name, the capacity.

Excessively long solutions that use more than 10 lines of code may lose points. A program that uses the function *sumAbsolute* follows.

```
int main() {
    double x[4] = { -1, -2, 3, 0};
    cout << sumAbsolute(x, 4) << endl; // prints 6 (this is found as 1 + 2 + 3 + 0).
    return 0;
}
```

Answer:

```
double sumAbsolute(double a[], int cap) {
    double answer = 0.0;
    for (int c = 0; c < cap; c++) {
        if (a[c] < 0) answer -= a[c];
        else answer += a[c];
    }
    return answer;
}
```

Problem 4 The recursive function `changeDigits` has 3 parameters `x`, `a`, `b`. The parameter `x` is a positive integer and the other parameters are single digit integers. The function considers the digits of `x` and returns a result obtained by changing any copy of `a` to become `b`. For example `changeDigits(1331, 1, 9)` returns 9339 and `changeDigits(1331, 1, 0)` returns 330.

An implementation of this function with parts of the code covered up is given below. There is also a main program that uses it.

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```
int changeDigits(PART (a)) {
    if ( PART (b) ) return b;
    if ( PART (c) ) return x;
    int y = PART (d);
    int z = changeDigits(PART (e));
    return PART (f);
}

int main() {
    cout << changeDigits(1331, 1, 2) << endl; // prints 2332
    cout << changeDigits(1331, 1, 0) << endl; // prints 330
    cout << changeDigits(1331, 2, 5) << endl; // prints 1331
    return 0;
}
```

(a) Give a replacement for PART (a) to declare the parameters `x`, `a`, `b` :

Answer: PART (a) is `int x, int a, int b`

(b) Give a replacement for PART (b) as a base case of recursion:

Answer: PART (b) is `x == a`

(c) Give a replacement for PART (c) as a second base case of recursion:

Answer: PART (c) is `x < 10`

(d) Give a replacement for PART (d) to change digits in `x / 10`:

Answer: PART (d) is `changeDigits(x / 10 , a, b)`

(e) Give a replacement for PART (e) to change the last digit, if necessary:

Answer: PART (e) is `x % 10, a, b`

(f) Give a replacement for PART (f) to return the answer:

Answer: PART (f) is `10 * y + z`

Solutions

10.10am – 11.00am, Wednesday, May 5, 2021

Problem 1 Write the best **title lines** for the functions that are called by the following main program. **Do not supply the blocks for the functions.**

```
int main() {
    int x = 12, y = 36;
    double z[4] = {1.1, -1.11, 2.5, 5.7};
    string s = "twelve", q = "Queens";

    // a. The function sameSign reports whether two numbers have the same sign. Here it returns false.
    if ( !sameSign(z[0], z[1]) ) cout << "Opposite signs\n"; // (a)

    // b. The function cutDuplicates removes all duplicate letters from a word.
    cutDuplicates( q ); // (b)
    cout << q << endl; // prints Quns after cutting the duplicated e

    // c. The function ratio returns x/y rounded to 3 decimal places eg as 0.333.
    cout << ratio(x, y) << endl; // (c)

    // d. The function translate returns the integer corresponding to the name of a number. Here 12.
    cout << translate(s) << endl; // (d)

    // e. A mystery function.
    mystery(mystery(sameSign(z[0], ratio(x,y)))); // (e)

    return 0;
}
```

(a) Title line for **sameSign** as called at the line marked (a).

Answer: `bool sameSign(double x, double y)`

(b) Title line for **cutDuplicates** as called at the line marked (b).

Answer: `void cutDuplicates(string &x)`

(c) Title line for **ratio** as called at the line marked (c).

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

(d) Title line for **translate** as called at the line marked (d).

Answer: `int translate(string s)`

(e) Title line for **mystery** as called at the line marked (e).

Answer: `bool mystery(bool b)`

Problem 2 Consider the following C++ program. The program makes use of a function `permuteArray` that moves the first entry in an array to the end. So that if the array has capacity 5 and entries 1, 2, 3, 4, 5 in this order, the function `permuteArray` moves the entries so that they are ordered as 2, 3, 4, 5, 1.

Make sure to use your own 8-digit CUNY ID number as the number entered as input to the program. It would be a very bad idea to give answers based on another student's ID number!

```
int main() {
    int id, a[10] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3};

    cout << "Enter your 8-digit CUNY id number: ";
    cin >> id;    // assume that the user types YOUR OWN CUNY ID number

    cout << id << endl;           // line (a)
    cout << a[ id % 10 ] << endl; // line (b)

    permuteArray(a, 10);
    cout << a[ a[0] ] << endl;    // line (c)
    cout << a[ 4 ] % a[ 5 ] << endl; // line (d)
    cout << a[ 0 ] + a[ 2 ] % a[ 3 ] << endl; // line (e)

    return 0;
}
```

Enter your 8-digit CUNY id number: (a) What is the output from the instruction beginning on line (a)?

Answer:

12345678

This answer is based on the ID number 12345678. Actual answers will be different.

(b) What is the output from the instruction beginning on line (b)?

Answer:

5

The answer will be the array entry indexed by the last digit of the answer to (a).

(c) What is the output from the instruction beginning on line (c)?

Answer:

4

(d) What is the output from the instruction beginning on line (d)?

Answer:

1

(e) What is the output from the instruction beginning on line (e)?

Answer:

2

Problem 3 Write a function called *numberNegative* that returns the number of negative entries in a 2-dimensional array of integers with 3 columns. The function should use 3 parameters as follows: the array name, the number of rows, the number of columns.

Excessively long solutions that use more than 10 lines of code may lose points. A program that uses the function *numberNegative* follows.

```
int main() {
    int x[4][3] = { {0,-1,2}, {-3,4,-5}, {6,-7,8}, {9,10,11}};
    cout << numberNegative(x, 4, 3) << endl; // prints 4 (because it counts -1,-3,-5,-7)
    return 0;
}
```

Answer:

```
int numberNegative(int a[][3], int rows, int cols) {
    int answer = 0;
    for (int r = 0; r < rows; r++)
        for (int c = 0; c < cols; c++)
            if (a[r][c] < 0) answer++;
    return answer;
}
```

Problem 4 The recursive function `swapDigits` has 3 parameters `x`, `a`, `b`. The parameter `x` is a positive integer and the other parameters are single digit integers. The function considers the digits of `x` and returns a result obtained by changing any copy of `a` to become `b` and any copy of `b` to become `a`. For example `swapDigits(13531, 1, 3)` returns 31513 and `swapDigits(1331, 1, 0)` returns 330.

An implementation of this function with parts of the code covered up is given below. There is also a main program that uses it.

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```
int swapDigits(PART (a)) {
    if ( x == a || x == b ) return a + b PART (b) x;
    if ( PART (c) ) return x;
    int y = PART (d);
    int z = swapDigits(PART (e));
    return PART (f);
}

int main() {
    cout << swapDigits(1331, 1, 3) << endl; // prints 3113
    cout << swapDigits(1331, 1, 0) << endl; // prints 330
    cout << swapDigits(1331, 2, 5) << endl; // prints 1331
    return 0;
}
```

(a) Give a replacement for PART (a) to declare the parameters `x`, `a`, `b` :

Answer: PART (a) is `int x, int a, int b`

(b) Give a replacement for PART (b) as an operation to calculate the answer in these base cases:

Answer: PART (b) is `-`

(c) Give a replacement for PART (c) as a second base case of recursion:

Answer: PART (c) is `x < 10`

(d) Give a replacement for PART (d) to change digits in `x / 10`:

Answer: PART (d) is `swapDigits(x / 10 , a, b)`

(e) Give a replacement for PART (e) to change the last digit, if necessary:

Answer: PART (e) is `x % 10, a, b`

(f) Give a replacement for PART (f) to return the answer:

Answer: PART (f) is `10 * y + z`