

**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() {
    int a[4] = {3, 31, 314, 3141};
    int a2[2][2] = {{3, 31}, {314, 3141}};
    int b = 3, c = 1;

    cout << min(b, 4) << endl;           // (a) prints: 3
    swap(b, c);                          // (b) swaps b and c
    a[0] = max(a, 4);                    // (c) sets a[0] to 3141
    cout << second(a2, 2, 2) << endl;     // (d) prints: 314
    makeZero(a2[1][1]);                  // (e) makes it 0
    return 0;
}
```

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

**Answer:**

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

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

**Answer:**

```
void swap(int &x, int &y)
```

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

**Answer:**

```
int max(int x[], int c)
```

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

**Answer:**

```
int second(int a[][2], int r, int c)
```

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

**Answer:**

```
void makeZero(int &x)
```

**Problem 2** Consider the following C++ program.

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

int up(int x[], int c) {
    if (c == 1) cout << x[1];
    if (c < 2) return 23;
    if (c == 2) return x[1];
    return x[c] + up(x, c - 1);
}

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

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

**Answer:**

4

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

**Answer:**

314159

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

**Answer:**

123

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

**Answer:**

14

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

**Answer:**

7

**Problem 3** Write a function called *averageOdd* that returns the average of all of the odd numbers in a 2-dimensional array with 3 columns. If no odd numbers are present, it should return a result of 0. Excessively long solutions that use more than 15 lines of code may lose points.

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

```
int main() {
    int data[2][3] = {{3, 1, 4},{2, 7, 1}};
    cout << averageOdd(data, 2, 3) << endl;    // prints 3.0
                                                // because the odd entries 3, 1, 7, 1 average to 3.0
    return 0;
}
```

**Answer:**

```
double averageOdd(int a[][3], int r, int c) {
    int sum = 0, count = 0;
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            if (a[i][j] % 2 != 0) {
                sum += a[i][j];
                count++;
            }
    if (count == 0) return 0;
    return sum / ((double) count);
}
```

**Problem 4** Write a function called *interlaceDigits* that uses two positive integer parameters with the same number of digits and returns an integer that begins with the first digit of the first parameter, then the first digit of the second parameter, then the second digits of the parameters, and so on until all digits are used. If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 10 lines of code may lose points.

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

```
int main() {
    cout << interlaceDigits(1, 2) << endl;           // prints 12
    cout << interlaceDigits(117, 302) << endl;       // prints 131072
    cout << interlaceDigits(1357, 2468) << endl;     // prints 12345678
    return 0;
}
```

**Answer:**

```
int interlaceDigits(int x, int y) {
    if (x == 0) return 0;
    return 100 * interlaceDigits(x / 10, y / 10) + 10 * (x % 10) + y % 10;
}
```

Solutions

09.00am – 09.50am, Wednesday, November 29, 2017

**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() {
    int i = 123, arr1 [3] = {1, 2, 3}, arr2 [2][2] = {{1, 0}, {2, 4}};
    double d1 = 1.23, d2 = 12.3;
    printLine (arr2, 2, 2);           // (a) prints: 1 0 2 4
    printFancy (arr1, 3);             // (b) prints: 1 * 2 ** 3 ***
    cout << doNothing (i, (int) d1); // (c) prints: This is a useless function
    switchValues (d1, d2); // (d) switches the values: now, d1 = 12.3, d2 = 1.23
    goodDayWishes ();                 // (e) prints: Have a good day
    return 0;
}
```

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

**Answer:**

```
void printLine (int array [] [2], int rowCap, int colCap)
```

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

**Answer:**

```
void printFancy (int array [], int capacity)
```

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

**Answer:**

```
string doNothing (int i1, int i2)
```

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

**Answer:**

```
void switchValues (double &d1, double &d2)
```

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

**Answer:**

```
void goodDayWishes ()
```

**Problem 2** Consider the following C++ program.

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

int up(int x[], int c) {
    if (c == 1) cout << x[1];
    if (c < 2) return 47;
    if (c == 2) return x[1];
    return x[c] + up(x, c - 1);
}

int main() {
    int x[6] = {2, 7, 1, 8, 2, 8};
    cout << 3 + x[1] << endl;           // line (a)
    for (int i = 0; i < 6; i++) cout << x[i];    cout << endl; // line (b)
    cout << up(x, 1) << endl;           // line (c)
    cout << up(x, 2) << x[2] << endl;    // line (d)
    cout << up(x, 4) << endl;           // line (e)
}
```

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

**Answer:**

10

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

**Answer:**

271828

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

**Answer:**

747

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

**Answer:**

71

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

**Answer:**

17

**Problem 3** Write a function called *numberNegative* that returns the number of negative elements in a 2-dimensional array with 3 columns. Excessively long solutions that use more than 12 lines of code may lose points.

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

```
int main() {
    double data[2][3] = {{-3.0, 1, 4.5},{-2.2, 7, 1.4}};
    cout << numberNegative(data, 2, 3) << endl;    // prints 2
                                                    // because there are 2 negatives -3.0 and -2.2
    return 0;
}
```

**Answer:**

```
int numberNegative(double a[][3], int r, int c) {
    int count = 0;
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            if (a[i][j] < 0) count++;
    return count;
}
```

**Problem 4** Write a function called *interweaveDigits* that uses two positive integer parameters with the same number of digits and returns an integer that begins with the first digit of the second parameter, then the first digit of the first parameter, then the second digits of the parameters, and so on until all digits are used. If a negative parameter is given, or if parameters with unequal numbers of digits are given your function can return any result of your choosing. Excessively long solutions that use more than 10 lines of code may lose points.

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

```
int main() {
    cout << interweaveDigits(2, 1) << endl;           // prints 12
    cout << interweaveDigits(302, 117) << endl;       // prints 131072
    cout << interweaveDigits(2468, 1357) << endl;     // prints 12345678
    return 0;
}
```

**Answer:**

```
int interweaveDigits(int x, int y) {
    if (x == 0) return 0;
    return 100 * interweaveDigits(x / 10, y / 10) + 10 * (y % 10) + x % 10;
}
```



Solutions

02.45pm – 03.35pm, Wednesday, November 29, 2017

**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() {
    int a[4] = {3, 31, 314, 3141};
    int a2[2][2] = {{3, 31}, {314, 3141}};
    int b = 3, c = 1;

    cout << min(b, 4) << endl;           // (a) prints: 3
    swap(b, c);                          // (b) swaps b and c
    a[0] = max(a, 4);                    // (c) sets a[0] to 3141
    cout << second(a2, 2, 2) << endl;     // (d) prints: 314
    makeZero(a2[1][1]);                 // (e) makes it 0
    return 0;
}
```

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

**Answer:**

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

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

**Answer:**

```
void swap(int &x, int &y)
```

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

**Answer:**

```
int max(int x[], int c)
```

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

**Answer:**

```
int second(int a[][2], int r, int c)
```

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

**Answer:**

```
void makeZero(int &x)
```

**Problem 2** Consider the following C++ program.

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

void up(int x[][3], int rows, int cols) {
    for (int c = 0; c < cols; c++) for (int r = 0; r < rows; r++)
        cout << 10 + x[r][c];
    cout << endl;
}

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

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

**Answer:**

5

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

**Answer:**

5

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

**Answer:**

2

6

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

**Answer:**

13

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

**Answer:**

13111115

**Problem 3** Write a function called *sum3* that returns the sum of all of the 3-digit numbers in an array. Excessively long solutions that use more than 12 lines of code may lose points.

For example, a program that uses the function *sum3* follows.

```
int main() {
    int x[6] = {3, 31, 314, 111, 4000, 100};
    cout << sum3(x, 6) << endl;    // prints 525
    // because the 3-digit numbers 314, 111, 100 add to 525
    return 0;
}
```

**Answer:**

```
int sum3(int a[], int c) {
    int sum = 0;
    for (int i = 0; i < c; i++)
        if (99 < a[i] && a[i] < 1000)
            sum += a[i];
    return sum;
}
```

**Problem 4** Write a function called *gcb* that uses two positive integer parameters and returns the greatest common beginning to the two numbers. For example, the greatest common beginning of 1235 and 1248 is 12. If the two parameters begin differently the function should return 0. If a negative parameter is given your function can return any result of your choosing. Excessively long solutions that use more than 10 lines of code may lose points.

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

```
int main() {
    cout << gcb(123, 223) << endl;           // prints 0
    cout << gcb(117, 119) << endl;           // prints 11
    cout << gcb(1357, 136578) << endl;       // prints 13
    return 0;
}
```

**Answer:**

```
int gcb(int x, int y) {
    if (x == y) return x;
    if (x > y) return gcb(x/10, y);
    return gcb(x, y/10);
}
```

Solutions

02.45pm – 03.35pm, Wednesday, November 29, 2017

**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() {
    int i = 123, arr1 [3] = {1, 2, 3}, arr2 [2][2] = {{1, 0}, {2, 4}};
    double d1 = 1.23, d2 = 12.3;
    printLine (arr2, 2, 2);           // (a) prints: 1 0 2 4
    printFancy (arr1, 3);             // (b) prints: 1 * 2 ** 3 ***
    cout << doNothing (i, (int) d1); // (c) prints: This is a useless function
    switchValues (d1, d2); // (d) switches the values: now, d1 = 12.3, d2 = 1.23
    goodDayWishes ();                 // (e) prints: Have a good day
    return 0;
}
```

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

**Answer:**

```
void printLine (int array [] [2], int rowCap, int colCap)
```

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

**Answer:**

```
void printFancy (int array [], int capacity)
```

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

**Answer:**

```
string doNothing (int i1, int i2)
```

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

**Answer:**

```
void switchValues (double &d1, double &d2)
```

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

**Answer:**

```
void goodDayWishes ()
```

**Problem 2** Consider the following C++ program.

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

void up(int x[][3], int rows, int cols) {
    for (int c = 0; c < cols; c++) for (int r = 0; r < rows; r++)
        cout << x[r][c] - 7;
    cout << endl;
}

int main() {
    int x[3][3] = {{2, 7, 1}, {8, 2, 8}, {1, 8, 2}};
    cout << x[2][2] << endl;           // line (a)
    cout << x[x[2][0]][x[2][0]] << endl; // line (b)
    for (int r = 0; r < 2; r++) cout << x[2][r] << endl; // line (c)
    up(x, 1, 1);                       // line (d)
    up(x, 2, 2);                       // line (e)
}
```

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

**Answer:**

2

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

**Answer:**

2

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

**Answer:**

1  
8

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

**Answer:**

-5

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

**Answer:**

-510-5

**Problem 3** Write a function called *numberFreddy* that returns the number of entries of an array equal to "Freddy". Excessively long solutions that use more than 12 lines of code may lose points.

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

```
int main() {
    string data[5] = {"Kelly", "Jack", "Freddy", "Arthur", "Freddy"};
    cout << numberFreddy(data, 5) << endl;    // prints 2
    return 0;
}
```

**Answer:**

```
int numberFreddy(string a[], int c) {
    int count = 0;
    for (int i = 0; i < c; i++)
        if (a[i] == "Freddy") count++;
    return count;
}
```

**Problem 4** Write a function called *gce* that uses two positive integer parameters and returns the greatest common ending to the two numbers. For example, the greatest common ending of 1234 and 134 is 34. If the two parameters end differently the function should return 0. If a negative parameter is given your function can return any result of your choosing. Excessively long solutions that use more than 10 lines of code may lose points.

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

```
int main() {
    cout << gce(123, 123) << endl;          // prints 123
    cout << gce(123, 223) << endl;          // prints 23
    cout << gce(117, 119) << endl;          // prints 0
    cout << gce(1357, 13657) << endl;       // prints 57
    return 0;
}
```

**Answer:**

```
int gce(int x, int y) {
    if (x == 0) return 0;
    if (x % 10 != y % 10) return 0;
    return 10 * gce(x / 10, y / 10) + x % 10;
}
```