

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() {
    string s; char c = 'A'; double d = 1.1;
    int a[4] = {3, 1, 4, 2};
    bool b[2][3] = {{true, false, true}, {false, true, true}};

    d = randomNumber(); cout << d << endl; // (a) prints a random number eg 1.5
    printThem(a, 4);                        // (b) prints 3142
    b[1][0] = majority(b, 2, 3); if (b[1][0]) cout << "true\n"; // (c) prints true
    doubleIt(a[1]); cout << a[1] << endl;    // (d) prints: 2
    s = asString(b[0][0]); cout << s << endl; // (e) prints: True
    return 0;
}
```

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

Answer:

```
double randomNumber()
```

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

Answer:

```
void printThem(int x[], int cap)
```

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

Answer:

```
bool majority(bool x[][3], int r, int c)
```

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

Answer:

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

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

Answer:

```
string asString(bool x)
```

Problem 2 Consider the following C++ program.

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

double down(int x[], int cap, int gap) {
    double ans = 0.0;
    for (int i = 0; i < cap; i+= gap)
        ans += x[i];
    return ans / 10;
}
```

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

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

Answer:

4

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

Answer:

1

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

Answer:

0.9

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

Answer:

0.4

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

Answer:

0.8

Problem 3 Write a function called *diff2* that returns the absolute value of the difference of the first two digits of a positive integer parameter. If the parameter has just one digit, that digit should be returned.

For example, a program that uses the function *diff2* follows.

```
int main() {
    cout << diff2(7070);           // prints 7
    cout << endl;
    cout << diff2(7907);           // prints 2
    cout << endl;
    cout << diff2(7);              // prints 7
    cout << endl;
    return 0;
}
```

Answer:

```
int diff2(int x) {
    if (x < 100) {
        int ans = x / 10 - x % 10;
        if (ans < 0) ans = -ans;
        return ans;
    }
    return diff2(x / 10);
}
```

Problem 4 Write a function called *indexFirstOdd* that returns the index of the first odd valued entry in an array or returns -1 if there is no odd value. (The index of an entry is its position in the array.)

For example,

```
int main() {
    int x[3] = {8, 8, 7};
    int y[5] = {7, 2, 5, 1, 9};
    int z[2] = {2, 2};
    cout << indexFirstOdd(x, 3) << endl;      // prints 2
    cout << indexFirstOdd(y, 5) << endl;      // prints 0
    cout << indexFirstOdd(z, 2) << endl;      // prints -1
    return 0;
}
```

Answer:

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

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() {
    string s; char c = 'A'; double d = 1.1;
    int a[4] = {3, 1, 4, 2};
    bool b[2][3] = {{true, false, true}, {false, true, true}};

    s = asString(c); cout << s << endl;      // (a) prints: A
    doubleIt(d); cout << d << endl;          // (b) prints: 2.2
    doubleThem(a, 4); cout << a[0] << endl;  // (c) prints 6
    printArray(b, 2, 3);                     // (d) prints TFT FTT
    c = randomLetter(); cout << c << endl;   // (e) prints a random letter eg Z
    return 0;
}
```

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

Answer:

```
string asString(char x)
```

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

Answer:

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

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

Answer:

```
void doubleThem(int x[], int cap)
```

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

Answer:

```
void printArray(bool x[][3], int r, int c)
```

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

Answer:

```
char randomLetter()
```

Problem 2 Consider the following C++ program.

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

double down(int x[], int cap, int &gap) {
    double ans = 0.0;
    for (int i = 0; i < cap; i+= gap)
        ans += x[i];
    gap += 2;
    return ans / 10;
}
```

```

}

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

```

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

Answer:

1

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

Answer:

1.4

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

Answer:

1.1

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

Answer:

0.4

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

Answer:

1

Problem 3 Write a function called *unlucky* that returns an answer of *true* if the first two digits of a positive integer parameter add to 13. Otherwise it returns *false*. (It returns *false* if the parameter has fewer than 2 digits.) For example, a program that uses the function *unlucky* follows.

```

int main() {
    if (unlucky(6789)) cout << "Unlucky!\n"; // prints Unlucky!
    if (unlucky(6889)) cout << "Unlucky!\n"; // prints
    if (unlucky(6)) cout << "Unlucky!\n"; // prints
    if (unlucky(49)) cout << "Unlucky!\n"; // prints Unlucky!
    return 0;
}

```

Answer:

```

bool unlucky(int x) {
    if (x < 100)
        return x / 10 + x % 10 == 13;
    return unlucky(x / 10);
}

```

Problem 4 Write a function called *lastOdd* that returns the last odd valued entry in an array or returns 0 if there is no odd value.

For example,

```
int main() {
    int x[3] = {8, 1, 7};
    int y[5] = {1, 2, 5, 4, 6};
    int z[2] = {2, 2};
    cout << lastOdd(x, 3) << endl;      // prints 7
    cout << lastOdd(y, 5) << endl;      // prints 5
    cout << lastOdd(z, 2) << endl;      // prints 0
    return 0;
}
```

Answer:

```
int lastOdd(int a[], int cap) {
    for (int i = cap - 1; i >= 0; i--)
        if ((a[i] % 2) == 1) return a[i];
    return 0;
}
```

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() {
    string s; char c = 'A'; double d = 4.0;
    int a[4] = {3, 1, 4, 2};
    bool b[2][3] = {{true, false, true}, {false, true, true}};

    printThem(b, 2, 3);                // (a) prints TFT FTT
    fixLies(b, 2, 3); printThem(b, 2, 3); // (b) prints FTF TFF
    d = cubeIt(d); cout << d << endl;    // (c) prints: 64.0
    cubeInt(a[2]); cout << a[2] << endl;  // (d) prints: 64
    a[1] = reverseDigits(a[2]); cout << a[1] << endl; // (e) prints: 1
    return 0;
}
```

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

Answer:

```
void printThem(bool x[][3], int r, int c)
```

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

Answer:

```
void fixLies(bool x[][3], int r, int c)
```

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

Answer:

```
double cubeIt(double x)
```

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

Answer:

```
void cubeInt(int &x)
```

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

Answer:

```
int reverseDigits(int x)
```

Problem 2 Consider the following C++ program.

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

double down(int x[], int cap, int &gap) {
    double ans = 0.0;
    for (int i = 0; i < cap; i+= gap)
        ans += x[i];
    gap += 2;
    return ans / 10;
}
```

```

}

int main() {
    int x[4] = {9, 1, 3, 2};
    int a = 4, b = 2;
    cout << x[9/3] << endl;           // line (a)
    cout << down(x, a, b) << endl;     // line (b)
    cout << down(x, a, b) << endl;     // line (c)
    cout << down(x, x[2], x[x[2]]) << endl; // line (d)
    cout << x[3] << endl;             // line (e)
}

```

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

Answer:

2

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

Answer:

1.2

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

Answer:

0.9

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

Answer:

1.2

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

Answer:

4

Problem 3 Write a function called *add7* that returns an answer found by putting a 7 in front of the first digit of a positive integer.

For example, a program that uses the function *add7* follows.

```

int main() {
    cout << add7(1) << "\n";    // prints 71
    cout << add7(17) << "\n";   // prints 717
    cout << add7(456) << "\n";  // prints 7456
    return 0;
}

```

Answer:

```

int add7(int x) {
    if (x == 0) return 7;
    return add7( x / 10) * 10 + x % 10;
}

```

Problem 4 Write a function called *evenLessOdd* that returns the sum of the even valued entries minus the sum of the odd valued entries in an array of integers.

For example, a program that uses the function *evenLessOdd* follows. The first output is $2 = 8 - 1 - 5$ and the second is $-10 = -1 - 1 - 5 - 3$.


```
int main() {
    int x[3] = {8, 1, 5};
    int y[4] = {1, 1, 5, 3};
    cout << evenLessOdd(x, 3) << endl;        // prints 2
    cout << evenLessOdd(y, 4) << endl;        // prints -10
    return 0;
}
```

Answer:

```
int evenLessOdd(int a[], int cap) {
    int answer = 0;
    for (int i = 0; i < cap; i++)
        if ((a[i] % 2) == 0) answer += a[i];
        else answer -= a[i];
    return answer;
}
```