

Solutions

08.30am – 10.30am, Monday, December 19, 2016

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 x = 2, y = 3, z[4];
    bool a = true, b = false, c[4];
    string s = "Hello", t = "goodbye", u[4][5];
    for (int i = 0; i < 4; i++) c[i] = data(x, y, 2.5);    // (a)
    setToFive(z, c, 4); cout << z[1] << endl;           // (b) prints 5
    y = speedLimit(x, z[1]); cout << x << y << endl;      // (c) prints 55
    cout << numberStrings(4, u, 5) << endl;               // (d) prints 20
    f(numberStrings(0, u, 0), data(y, x, f(20, a || b))); // (e)
    return 0;
}
```

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

Answer:

```
bool data(int x, int y, double z)
```

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

Answer:

```
void setToFive(int a[], bool b[], int cap)
```

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

Answer:

```
int speedLimit(int &a, int b)
```

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

Answer:

```
int numberStrings(int a, string b[][5], int c)
```

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

Answer:

```
double f(int a, bool b)
```

Problem 2 Consider the following C++ program.

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

int F(int x[], int c) {
    if (c < 3) return 0;
    return x[c - 1] + F(x, c - 1);
}

int G(int a, int &b) {
    b = b - a;
    a = b + a;
    return a;
}

int main() {
    int a = 4, b = 1;
    int x[5] = {3, 1, 4, 1, 5};
    string s = "Problem Number 2";
    cout << x[2 + 2] + x[2] << endl;           // line (a)
    cout << s.substr(2, 3) << endl;             // line (b)
    cout << s.substr(s.find("b")) << endl;       // line (c)
    cout << G(b, a); cout << a << b << endl;    // line (d)
    cout << F(x, 5) << endl;                   // line (e)
    return 0;
}
```

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

Answer:

9

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

Answer:

obl

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

Answer:

blem Number 2

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

Answer:

431

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

Answer:

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][4] = {{1, 2, 3, 4}, {0, 1, 2, 3}};
    int b[4] = {3, 1, 4, 1};
    int x = 1, y = 2;
    string s = "hello";
    // (a) Return true if at least one of x and y is positive. Here Yes is printed
    if (positive(x, y)) cout << "Yes" << endl;
    // (b) Return the sum of the first row. Here 10 is printed.
    cout << rowSum(a, 2, 4) << endl;
    // (c) Return the smallest element. Here 1 is printed.
    cout << smallest(b, 4) << endl;
    // (d) Remove the first letter. Here ello is printed.
    cout << removeFirst(s) << endl;
    // (e) Insert an X at the specified position. Here heXllo is printed.
    addX(s, 2);
    cout << s << endl;
    return 0;
}
```

Answer:

(a)

```
bool positive(int x, int y) {
    return x > 0 || y > 0;
}
```

(b)

```
int rowSum(int a[][4], int r, int c) {
    int ans = 0;
    for (int j = 0; j < c; j++)
        ans += a[r][j];
    return ans;
}
```

(c)

```
int smallest(int x[], int c) {
    int ans = x[0];
    for (int i = 0; i < c; i++)
        if (x[i] < ans) ans = x[i];
    return ans;
}
```

(d)

```
string removeFirst(string s) {
    return s.substr(1);
}
```

(e)

```
void addX(string &s, int y) {
    s.insert(y, "X");
}
```

Problem 4 Write a function called *randFill* that fills the entries of an array with random integers in the range from 10 to 99 (inclusive). (You should use the *rand* function to generate the values. You do not need to call *srand*. Your solution should use no more than 6 lines of code.)

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

```
int main() {
    int x[5];
    randFill(x, 5);
    for (int i = 0; i < 5; i++)
        cout << x[i] << " "; // prints 5 random numbers
    cout << endl;           // such as 93 73 12 69 40
    return 0;
}
```

Answer:

```
#include <cstdlib>

void randFill(int x[], int cap) {
    for (int i = 0; i < cap; i++)
        x[i] = rand() % 90 + 10;
}
```

Problem 5 Write a function called *evenUp* that returns the result of increasing the first even digit in a positive integer parameter by 1. (Your solution should use no more than 10 lines of code. Your function can return any convenient value of your choice if the parameter is not positive.)

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

```
int main() {
    cout << evenUp(1232) << endl; // prints 1332  only the first even 2 changes
    cout << evenUp(1332) << endl; // prints 1333
    cout << evenUp(1333) << endl; // prints 1333  no even digit to change
    cout << evenUp(22) << endl;   // prints 32
    cout << evenUp(2) << endl;    // prints 3
    return 0;
}
```

Answer:

```
int evenUp(int x) {
    if (x <= 0) return 0;
    int y = evenUp(x / 10);
    if (x % 2 == 1) return 10 * y + x % 10;
    if (y > x / 10) return 10 * y + x % 10;
    return x + 1;
}
```

Problem 6 Write a complete C++ program that is to be used for a psychology study into random number choices by a human volunteer. Your program is to operate as follows. (Programs that correctly carry out some of the tasks will receive partial credit. Your program should not be more than 30 lines long.)

Ask the user (the volunteer) to repeatedly type 2 digit numbers onto the screen.

Read the user input and discard any number that is less than 10 or greater than 99, but keep track of numbers within this range.

When the total of the legal numbers typed exceeds 100000 the experiment ends and the program prints a summary with the following form (with one line of output for each of the numbers from 10 to 99):

User chose 99 for 2.1% of choices.

User chose 98 for 0.7% of choices.

User chose 97 for ...

Answer:

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

int main() {
    int n = 0, total = 0, x;
    int counts[100];
    for (int i = 0; i < 100; i++)
        counts[i] = 0;
    while (total <= 100000) {
        cout << "Enter a 2 digit number: ";
        cin >> x;
        if (10 <= x && x <= 99) {
            counts[x]++;
            total += x;
            n++;
        }
    }
    for (x = 99; x >= 10; x--)
        cout << "User chose " << x << " for "
            << 100.0 * counts[x] / n << "% of choices.\n";
}
```

Solutions

08.30am – 10.30am, Monday, December 19, 2016

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 = 2, b = 3, c[4];
    bool s = true, t = false, u[4];
    string x = "Hello", y = "goodbye", z[4][5];
    for (int i = 0; i < 4; i++) c[i] = data(x, y, 2.5); // (a)
    setToFive(z, c, 4); cout << z[1][1] << endl; // (b) prints 5
    y = speedLimit(x, z[1][1]); cout << x << y << endl; // (c) prints 55
    cout << numberStrings(s, t, b, u) << endl; // (d) prints 20
    numberStrings(f(a), f(a), a, u); // (e)
    return 0;
}
```

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

Answer:

```
int data(string x, string y, double z)
```

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

Answer:

```
void setToFive(string a[][5], int b[], int cap)
```

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

Answer:

```
string speedLimit(string &a, string b)
```

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

Answer:

```
int numberStrings(bool a, bool b, int c, bool d[])
```

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

Answer:

```
bool f(int a)
```

Problem 2 Consider the following C++ program.

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

int F(int x[], int c) {
    if (c < 1) return 0;
    return x[c - 1] + F(x, c - 1);
}

int G(int &a, int b) {
    b = b - a;
    a = b + a;
    return a;
}

int main() {
    int a = 7, b = 5;
    int x[5] = {3, 1, 4, 1, 5};
    string s = "String Question";
    cout << x[2 / 2] * x[2] << endl;           // line (a)
    cout << s.substr(2, 3) << endl;             // line (b)
    cout << s.substr(s.rfind("s")) << endl;     // line (c)
    cout << G(b, a); cout << a << b << endl;    // line (d)
    cout << F(x, 4) << endl;                   // line (e)
    return 0;
}
```

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

Answer:

4

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

Answer:

rin

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

Answer:

stion

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

Answer:

777

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

Answer:

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][4] = {{1, 2, 3, 4}, {0, 1, 2, 3}};
    int b[4] = {3, 1, 4, 1};
    int x = 1, y = 2;
    string s = "hello";
    // (a) Return true if both of x and y are positive. Here Yes is printed
    if (positive(x, y)) cout << "Yes" << endl;
    // (b) Return the sum of the second row. Here 6 is printed.
    cout << rowSum(a, 2, 4) << endl;
    // (c) Return the largest element. Here 4 is printed.
    cout << largest(b, 4) << endl;
    // (d) Return the first two letters. Here he is printed.
    cout << firstTwo(s) << endl;
    // (e) Insert a specified number of X's at the end. Here helloXX is printed.
    addX(s, 2);
    cout << s << endl;
    return 0;
}
```

Answer:

(a)

```
bool positive(int x, int y) {
    return x > 0 && y > 0;
}
```

(b)

```
int rowSum(int a[][4], int r, int c) {
    int ans = 0;
    for (int j = 0; j < c; j++)
        ans += a[r][j];
    return ans;
}
```

(c)

```
int largest(int x[], int c) {
    int ans = x[0];
    for (int i = 0; i < c; i++)
        if (x[i] > ans) ans = x[i];
    return ans;
}
```

(d)

```
string firstTwo(string s) {
    return s.substr(0, 2);
}
```

(e)

```
void addX(string &s, int y) {
    for (int i = 0; i < y; i++) s = s + "X";
}
```

Problem 4 Write a function called *randAdd* that changes each entry of an array by generating a random integer between 1 and 10 and adding it to the entry. (You should use the *rand* function to generate the values. You do not need to call *srand*. Your solution should use no more than 6 lines of code.)

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

```
int main() {
    int x[5] = {3, 1, 4, 1, 5};
    randAdd(x, 5);
    for (int i = 0; i < 5; i++)
        cout << x[i] << " "; // prints 5 randomly adjusted entries
    cout << endl;           // such as 93 73 12 69 40
    return 0;
}
```

Answer:

```
#include <cstdlib>

void randAdd(int x[], int cap) {
    for (int i = 0; i < cap; i++)
        x[i] += rand() % 10 + 1;
}
```

Problem 5 Write a function called *oddDown* that returns the result of decreasing the first odd digit in a positive integer parameter by 1. (Your solution should use no more than 10 lines of code. Your function can return any convenient value of your choice if the parameter is not positive.)

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

```
int main() {
    cout << oddDown(321) << endl; // prints 221 only the first odd digit changes
    cout << oddDown(221) << endl; // prints 220
    cout << oddDown(220) << endl; // prints 220 because no odd digit to decrease
    cout << oddDown(7) << endl;   // prints 6
    cout << oddDown(6) << endl;   // prints 6
    return 0;
}
```

Answer:

```
int oddDown(int x) {
    if (x <= 0) return 0;
    int y = oddDown(x / 10);
    if (x % 2 == 0) return 10 * y + x % 10;
    if (y < x / 10) return 10 * y + x % 10;
    return x - 1;
}
```

Problem 6 Write a complete C++ program that is to be used for a psychology study into random number choices by a human volunteer. Your program is to operate as follows. (Programs that correctly carry out some of the tasks will receive partial credit. Your program should not be more than 30 lines long.)

Ask the user (the volunteer) to repeatedly type single digit numbers onto the screen.

Read the user input and discard any number that is less than 1 or greater than 9, but keep track of numbers within this range.

When the total of the legal numbers typed exceeds 10000 the experiment ends and the program prints a list of the most frequent choice (or choices if two or more numbers are tied).

Output should appear as:

```
The most frequent choice(s): 3 7
```

Answer:

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

int main() {
    int total = 0, x, max = 0;
    int counts[10];
    for (int i = 0; i < 10; i++)
        counts[i] = 0;
    while (total <= 10000) {
        cout << "Enter a single digit number: ";
        cin >> x;
        if (1 <= x && x <= 9) {
            counts[x]++;
            total += x;
        }
    }
    for (x = 1; x <= 10; x++)
        if (counts[x] > max) max = counts[x];
    cout << "The most frequent choice(s): ";
    for (x = 1; x <= 10; x++)
        if (counts[x] == max) cout << x << " ";
    cout << endl;
}
```

Solutions

01.45pm – 03.45pm, Monday, December 19, 2016

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 x = 2, z[3] = {3, 1, 4};
    bool a = true, c[5];
    string s = "Hello", u[7][9];
    for (int i = 0; i < 4; i++) c[i] = A(x, x, 2.5);    // (a)
    cout << B(c, c, u);                                // (b) prints: part B
    x = C(x, u[1][1]); cout << x << endl;              // (c) prints 55
    D(4, z, 5); cout << z[1] << endl;                  // (d) prints 3
    E(E(a, s), s); cout << endl;                        // (e) prints 33
    return 0;
}
```

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

Answer:

```
bool A(int x, int y, double z)
```

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

Answer:

```
string B(bool a[], bool b[], string[][9])
```

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

Answer:

```
int C(int a, string b)
```

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

Answer:

```
void D(int a, int b[], int c)
```

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

Answer:

```
bool E(bool a, string b)
```

Problem 2 Consider the following C++ program.

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

int F(int a, int &b) {
    b = b - a;
    a = b + a;
    return a;
}

int G(int x[], int c) {
    if (c < 3) return 0;
    return x[c - 1] + G(x, c - 1);
}

int main() {
    int a = 5, b = 3;
    int x[5] = {2, 7, 1, 8, 2};
    string s = "Final Exam";
    cout << x[2 + 2] + x[2] << endl;           // line (a)
    cout << s.substr(2, 3) << endl;             // line (b)
    cout << s.substr(s.find("a")) << endl;       // line (c)
    cout << F(b, a); cout << a << b << endl;    // line (d)
    cout << G(x, 5) << endl;                   // line (e)
    return 0;
}
```

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

Answer:

3

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

Answer:

nal

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

Answer:

al Exam

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

Answer:

523

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

Answer:

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 x = 1, y = 2;
    string a[2][3] = {"CS", "111", "Final"}, {"Question", "number", "3"};
    string b[3] = {"An", "Easy", "Problem"};
    // (a) Return true if at least one of x and y is negative. Here nothing is printed
    if (negative(x, y)) cout << "Yes" << endl;
    // (b) Return the first entry in the first row. Here CS is printed.
    cout << firstEntry(a, 2, 3) << endl;
    // (c) Return the longest element. Here Problem is printed.
    cout << longest(b, 3) << endl;
    // (d) Remove the first letter. Here umber is printed.
    cout << removeFirst(a[1][1]) << endl;
    // (e) Insert a Q at the specified position of a string. Here CQS is printed.
    addQ(a[0][0], 1);
    cout << a[0][0] << endl;
    return 0;
}
```

Answer:

(a)

```
bool negative(int x, int y) {
    return x < 0 || y < 0;
}
```

(b)

```
string firstEntry(string a[][3], int r, int c) {
    return a[0][0];
}
```

(c)

```
string longest(string x[], int c) {
    string ans = x[0];
    for (int i = 0; i < c; i++)
        if (x[i].length() > ans.length()) ans = x[i];
    return ans;
}
```

(d)

```
string removeFirst(string s) {
    return s.substr(1);
}
```

(e)

```
void addQ(string &s, int y) {
    s.insert(y, "Q");
}
```

Problem 4 Write a function called *maxIndex* that reports the index of the row that contains the largest entry in a 2-dimensional array of integers (with 3 columns).

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

```
int main() {
    int x[3][3] = {{3,1,4},{1,5,9}, {2,6,5}};
    cout << maxIndex(x, 3, 3) << endl; // prints 1
                                         // because the entry 9 is in row 1
    return 0;
}
```

Answer:

```
int maxIndex(int x[][3], int r, int c) {
    int a = 0, b = 0;
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            if (x[i][j] > x[a][b]) {
                a = i;
                b = j;
            }
    return a;
}
```


Problem 5 Write a function called *evenUp* that returns the result of increasing the last even digit in a positive integer parameter by 1. (Your solution should use no more than 5 lines of code. Your function can return any convenient value of your choice if the parameter is not positive.)

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

```
int main() {
    cout << evenUp(1234) << endl; // prints 1235
    cout << evenUp(1335) << endl; // prints 1335
    cout << evenUp(2) << endl;    // prints 3
    cout << evenUp(3) << endl;    // prints 3
    return 0;
}
```

Answer:

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

Problem 6 Write a complete C++ program that is to be used for a marketing study into cent values that appear in gas prices. Your program is to operate as follows. (Programs that correctly carry out some of the tasks will receive partial credit. Your program should not be more than 30 lines long.)

Ask the user to repeatedly type numbers in the range 0 to 99 (representing cents in prices observed) onto the screen. Read the user input and discard any number that is out of range. As soon as every possible cent value has been seen at least once, the program ends by printing a summary with the following form (with one line of output for each of the numbers from 0 to 99):

```
99 cents for 12.1% of prices.  
98 cents for 0.7% of prices.  
97 cents for 0.35% of ...
```

Answer:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int numberOfValues = 0, total = 0, x;  
    int counts[100];  
    for (int i = 0; i < 100; i++)  
        counts[i] = 0;  
    while (numberOfValues < 100) {  
        cout << "Enter a number between 0 and 99: ";  
        cin >> x;  
        if (0 <= x && x <= 99) {  
            if (counts[x] == 0) numberOfValues++; // a new price has just been seen  
            counts[x]++;  
            total++;  
        }  
    }  
    for (x = 99; x >= 0; x--)  
        cout << x << " cents for " << 100.0 * counts[x] / total  
            << "% of prices.\n";  
}
```

Solutions

01.45pm – 03.45pm, Monday, December 19, 2016

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 = 2, c[3] = {3, 1, 4};  
    bool s = true, u[5];  
    string x = "Hello", z[7][9];  
    for (int i = 0; i < 4; i++) c[i] = A(x, x, 2.5);    // (a)  
    cout << B(c, c, u);                                // (b) prints: part B  
    x = C(x, u[1]); cout << x << endl;                // (c) prints 55  
    D(4, z, 5); cout << z[1][1] << endl;              // (d) prints 3  
    E(E(a, s), s); cout << endl;                      // (e) prints 33  
    return 0;  
}
```

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

Answer:

```
int A(string x, string y, double z)
```

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

Answer:

```
string B(int a[], int b[], bool[])
```

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

Answer:

```
string C(string a, bool b)
```

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

Answer:

```
void D(int a, string b[][9], int c)
```

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

Answer:

```
int E(int a, bool b)
```

Problem 2 Consider the following C++ program.

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

int F(int &a, int b) {
    b = b - a;
    a = b + a;
    return a;
}

int G(int x[], int c) {
    if (c < 1) return 0;
    return x[c - 1] + G(x, c - 1);
}

int main() {
    int a = 6, b = 4;
    int x[5] = {2, 7, 1, 8, 2};
    string s = "Queens College";
    cout << x[2 / 2] * x[2] << endl;           // line (a)
    cout << s.substr(5, 1) << endl;           // line (b)
    cout << s.substr(s.rfind("e")) << endl;    // line (c)
    cout << F(b, a); cout << a << b << endl; // line (d)
    cout << G(x, 4) << endl;                 // line (e)
    return 0;
}
```

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

Answer:

7

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

Answer:

s

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

Answer:

e

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

Answer:

666

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

Answer:

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 x = 1, y = 2;
    string a[2][3] = {"CS", "111", "Final"}, {"Question", "number", "3"};
    string b[3] = {"An", "Easy", "Problem"};
    // (a) Return true if both of x and y are negative. Here nothing is printed
    if (negative(x, y)) cout << "Yes" << endl;
    // (b) Return the first entry in the second column. Here 111 is printed.
    cout << firstEntry(a, 2, 3) << endl;
    // (c) Return the shortest element. Here An is printed.
    cout << shortest(b, 3) << endl;
    // (d) Return the first two letters. Here Fi is printed.
    cout << firstTwo(a[0][2]) << endl;
    // (e) Insert the specified number of Qs at the start of a string. Here QQCS is printed.
    addQ(a[0][0], 2);
    cout << a[0][0] << endl;
    return 0;
}
```

Answer:

(a)

```
bool negative(int x, int y) {
    return x < 0 && y < 0;
}
```

(b)

```
string firstEntry(string a[][3], int r, int c) {
    return a[r][1];
}
```

(c)

```
string shortest(string x[], int c) {
    string ans = x[0];
    for (int i = 0; i < c; i++)
        if (x[i].length() < ans.length()) ans = x[i];
    return ans;
}
```

(d)

```
string firstTwo(string s) {
    return s.substr(0, 2);
}
```

(e)

```
void addQ(string &s, int y) {
    for (int i = 0; i < y; i++)
        s = "Q" + s;
}
```

Problem 4 Write a function called *maxIndex* that reports the index of the column that contains the largest entry in a 2-dimensional array of integers (with 3 columns).

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

```
int main() {
    int x[3][3] = {{3,1,4},{1,5,9}, {2,6,5}};
    cout << maxIndex(x, 3, 3) << endl; // prints 2
                                         // because the entry 9 is in column 2
    return 0;
}
```

Answer:

```
int maxIndex(int x[][3], int r, int c) {
    int a = 0, b = 0;
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            if (x[i][j] > x[a][b]) {
                a = i;
                b = j;
            }
    return b;
}
```

Problem 5 Write a function called *oddDown* that returns the result of decreasing the last odd digit in a positive integer parameter by 1. (Your solution should use no more than 5 lines of code. Your function can return any convenient value of your choice if the parameter is not positive.)

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

```
int main() {
    cout << oddDown(3234) << endl; // prints 3224
    cout << oddDown(3224) << endl; // prints 2224
    cout << oddDown(1214) << endl; // prints 1204
    cout << oddDown(1204) << endl; // prints 204
    cout << oddDown(2) << endl;    // prints 2
    cout << oddDown(1) << endl;    // prints 0
    return 0;
}
```

Answer:

```
int oddDown(int x) {
    if (x % 2 == 1) return x - 1;
    if (x < 10) return x;
    return 10 * oddDown(x / 10) + x % 10;
}
```

Problem 6 Write a complete C++ program that is to be used for an economics study into mortgage interest rates. Your program is to operate as follows. (Programs that correctly carry out some of the tasks will receive partial credit. Your program should not be more than 30 lines long.)

Ask the user to repeatedly type integers in the range 0 to 8 (representing interest rates observed) onto the screen.

Read the user input and discard any number that is out of range. As soon as every possible input value has been seen at least once, the program ends by showing the most frequent rate (or rates in case of a tie). For example, output might be:

```
Most common rate(s): 3 4
```

Answer:

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

int main() {
    int numberOfValues = 0, x, max = 0;
    int counts[9];
    for (int i = 0; i < 9; i++)
        counts[i] = 0;
    while (numberOfValues < 9) {
        cout << "Enter a number between 0 and 8: ";
        cin >> x;
        if (0 <= x && x <= 8) {
            if (counts[x] == 0) numberOfValues++; // a new rate has just been seen
            counts[x]++;
        }
    }
    for (x = 0; x < 9; x++)
        if (counts[x] > max) max = counts[x];
    cout << "Most common rate(s): ";
    for (x = 0; x < 9; x++)
        if (counts[x] == max) cout << x << " ";
    cout << endl;
}
```