

Grades are available online from the Queens College webgrading system.

Problem 1 Write a complete C++ program that does the following.

1. It asks the user to enter some positive integers.
2. It reads positive integers from the user.
3. As soon as the user enters a non-positive integer, the program stops reading.
4. The program reports the sum of all the positive numbers that it read.

Here is an example of how the program should work:

```
Give me some positive integers:  1 12 1 100 -1000
sum: 114
```

Answer:

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

int main() {
    int sum = 0;
    int n = 1;
    cout << "Give me some positive integers: ";
    while (n > 0) {
        cin >> n;
        if (n > 0) sum += n;
    }
    cout << "sum: " << sum << endl;
    return 0;
}
```

Problem 2 Write header lines (prototypes) for the following functions. **Do not attempt to supply the blocks for the functions.**

- (a) A function called **isPrime** that tests whether an integer is prime. (For example, 7 is prime, but 9 is not.)

Answer:

```
bool isPrime(int x)
```

- (b) A function called **firstChar** which uses a string as input and returns the first character in the string.

Answer:

```
char firstChar(string x)
```

- (c) A function called **printThree** which uses as input an array of integers. The task of the function is to print the first three elements of the array.

Answer:

```
void printThree(int x[])
```

- (d) A function called **printChess** which uses as input an 8×8 array of characters that represents a chess board. The task of the function is to print the board to output.

Answer:

```
void printChess(char x[][8], int r, int c)
```

- (e) A function called **reverseWord** which is to use a string parameter and change it to become the string obtained by reversing its letters. (For example, an input string *was* would be changed to *saw*.)

Answer:

```
void reverseWord(string &x)
```

Problem 3 Write a function called *biggestEntry* that uses a two dimensional array (with 3 columns) with integer entries as its first parameter. It also uses parameters representing the row and column capacities. The function should return the value of the biggest entry in the array.

For example, a program that uses the function follows.

```
int main() {
    int x[2][3] = {{1,2,3},{4,7,3}};
    cout << biggestEntry(x, 2, 3) << endl;
    return 0;
}
```

It should print 7 (since 7 is the biggest entry in the array).

Answer:

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

Problem 4 Write C++ statements to carry out the following tasks. **Do not write complete programs**, just give a single line, or a few lines of C++ instructions. Assume that the following variables have been declared, and if necessary have values, for each part:

```
string f, l;
```

(i) Read a first name to *f* and a last name to *l*. Then, print out the string *f* followed by the string *l* on another line.

Answer:

```
cin >> f >> l;
cout << f << endl << l << endl;
```

(ii) Print the second character in *f*.

Answer:

```
cout << f[1];
```

(iii) Convert the string *f* to upper case letters and then print it.

Answer:

```
for (int i = 0; i < f.size(); i++) f[i] = toupper(f[i]);
cout << f;
```

(iv) Read a word into *f* from a user. If the program can find the smaller string "reddy" within the string *f*, print the word "Hello", otherwise do nothing.

Answer:

```
cin >> f;
if (f.find("reddy") >= 0) cout << "Hello";
```

(v) Print the last character of *l*.

Answer:

```
cout << l[l.size() - 1];
```

Problem 5 Consider the following C++ program.

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

void mystery(char x[][4], int a, int b, char k) {
    for (int r = a; r <= b; r++) for (int c = a; c <= b; c++)
        x[r][c] = k;
}

void print(char x[][4], int s) {
    for (int r = 0; r < s; r++) {
        for (int c = 0; c < s; c++) cout << x[r][c];
        cout << endl;
    }
    cout << endl;
}

int main() {
    char x[4][4];
    mystery(x, 0, 3, 'X'); print(x, 4);
    mystery(x, 1, 2, 'Y'); print(x, 4);
    mystery(x, 2, 3, 'Z'); print(x, 4);
    mystery(x, 3, 2, '0'); print(x, 4);
    return 0;
}
```

(a) What is the output from the first call to the function print?

Answer:

```
XXXX
XXXX
XXXX
XXXX
```

(b) What is the output from the second call to the function print?

Answer:

```
XXXX
XYYX
XYYX
XXXX
```

(c) What is the output from the third call to the function print?

Answer:

```
XXXX
XYYX
XYZZ
XXZZ
```

(d) What is the output from the fourth call to the function print?

Answer:

```
XXXX
XYYX
XYZZ
XXZZ
```

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 asks the user to enter a positive integer value, n .
2. The program reads a value entered by the user. If n is not positive, the program should exit.
3. It prints out the number of digits in n .
4. It prints the number digits in the binary representation of n .

Here is an example of how the program should work:

```
Enter a positive integer n: 17
Digits in n: 2
Binary digits in n: 5
```

The number of binary digits is 5 because the binary representation of 17 is 10001. However, it is not necessary for your program to determine this binary representation.

Answer:

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

int length(int x, int base) {
    if (x < base) return 1;
    return 1 + length(x / base, base);
}

int main() {
    int n;
    cout << "Enter a positive integer n: ";
    cin >> n;
    if (n <= 0) exit(1);
    cout << "Digits in n: " << length(n, 10) << endl;
    cout << "Binary digits in n: " << length(n, 2) << endl;
    return 0;
}
```