

**Problem 1** Write header lines (prototypes) for the following functions. Do not supply the blocks for the functions.

(a) A function called **sum** which returns the sum of 4 double precision values.

**Answer:**

(b) A function called **midDigit** that is used to return the middle digit of an integer.

**Answer:**

(c) A function called **isPositive** which is to return an answer of true if the sum of the entries of an array of double precision data is positive (and return false otherwise).

**Answer:**

(d) A function called **average2DArray** which is to print (to cout) the average of the entries in a 2-dimensional array (the array stores integers and has 10 rows and 15 columns).

**Answer:**

(e) A function called **makeZero** which is to use two integer input variables and change their values to zero. (After the function ends, the input variables must be zero.)

**Answer:**

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

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

void mystery(int n) {
    cout << n % 100;
    if (n < 1000) return;
    mystery(n/10);
}

main() {
    int x;
    cout << "Enter an integer: ";
    cin >> x;
    mystery(x);
    cout << endl;
    return 0;
}
```

What is the output from the program in response to the following user inputs.

(a) The user enters 5 for x.

(b) The user enters 512 for x.

(c) The user enters 4370 for x.

(d) The user enters 175560 for x.

**Problem 3** Write a function called *sum2D* that returns the sum of all elements in a 2-dimensional array that has 4 columns of integer entries.

For example, a program that uses the function *sum2D* follows.

```
int main() {
    int array[3][4] = {{1,2,3,4},{1,2,3,4},{1,2,3,4}};
    cout << sum2D(array, 3, 4) << endl;
    return 0;
}
```

The input values 3 and 4 specify the number of rows and columns in the array. The program should print an answer of 30 (since this is the sum of 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, and 4).

**Answer:**

**Problem 4** Write a complete C++ program that does the following.

1. It asks the user to enter a 5-digit integer value,  $n$ .
  2. The program reads a value entered by the user. If the value is not in the right range, the program should terminate.
  3. The program calculates and stores the 5 individual digits of  $n$ .
  4. The program outputs a “bar code” made of 5 lines of stars that represent the digits of the number  $n$ .
- For example, the following represents one run of the program. (The user chooses the number 16384.)

```
Enter a 5 digit integer:    16384
*
*****
***
*****
****
```

**Answer:**

**Problem 5** Write a complete C++ program that does the following.

1. It asks the user to enter 5 single digit positive integers.
2. If any number is out of range, it says: "That is too hard."
3. Otherwise it adds the numbers and prints their sum.

Here is an example of how the program should work:

```
Give me 5 single digit positive integers:    9 9 9 6 9
42
```

**Answer:**

**Problem 6** 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:

```
int x;
string f, l;
```

- (i) Read a user's first name to  $f$  and their last name to  $l$ .
- (ii) Print out the string  $f$  followed by the string  $l$  with a space between them.
- (iii) Set  $x$  to be  $1 - 2 + 3 - 4 + 5 - \dots + 999$ . The formula involves all integers from 1 to 999. Odd numbers are added, even numbers subtracted.
- (iv) Repeatedly double  $x$ , until the value of  $x$  exceeds 1024.
- (v) Read a word into  $f$  from a user. If the word is "Freddy", print output saying "Hello", otherwise do nothing.

**Problem 7** Consider the following C++ program.

```

#include <iostream>
using namespace std;

void mystery(string array[], int p[], int q) {
    if (q < 0) cout << "Help!" << endl;
    else if (q <= 2) cout << p[q] << endl;
    if (q > 2) {
        for (int i = 0; i <= q; i++) cout << array[p[i]] << " ";
        cout << endl;
    }
}

int main() {
    string x[5] = {"This", "is", "a", "dumb", "question"};
    int a[10] = {0, 4, 1, 3, 3, 3, 2, 2, 2, 2};
    mystery(x, a, -10);
    mystery(x, a, 0);
    mystery(x, a, 1);
    mystery(x, a, 3);
    mystery(x, a, 5);
    return 0;
}

```

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

**Answer:**

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

**Answer:**

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

**Answer:**

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

**Answer:**

(e) What is the output from the fifth call to the function `mystery`?

**Answer:**

**Problem 8** Write header lines (prototypes) for the following functions. **Do not attempt to supply the blocks for the functions.**

(a) A function called **isLeapYear** that tests whether an integer represents a leap year. (For example, 2008 is a leap year, but 2007 is not.)

**Answer:**

(b) A function called **temperatureDifference** which uses as input two double precision values that represent the temperature in New York measured in degrees Fahrenheit and the temperature in Paris measured in degrees Celsius. The function is to calculate and return the difference between the temperatures in degrees Fahrenheit.

**Answer:**

(c) A function called **addCurve** which uses as input an array of integer test scores. The task of the function is to add 10 to every score in the array.

**Answer:**

(d) A function called **printTicTacToe** which uses as input a  $3 \times 3$  array of characters that represents a Tic-Tac-Toe game. The task of the function is to print the board to output.

**Answer:**

(e) A function called **reverseDigits** which is to use an integer parameter and return the integer obtained by reversing the digits in the parameter.

**Answer:**

**Problem 9** Write a function called *biggestDigit* that uses an integer input parameter and returns the largest digit in the input. The input should be assumed to be positive.

For example, a program that uses the function follows.

```
int main() {
    cout << biggestDigit(1760) << endl;
    return 0;
}
```

It should print 7 (since 7 is the biggest digit in 1760).

A little extra credit will be given for good recursive solutions.

**Answer:**

**Problem 10** 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,  $n$  that is at most 100.
2. The program reads a value entered by the user. If  $n$  is not positive, or  $n$  is greater than 100, the program should exit.
3. It prints out all numbers between 1 and 1000 for which the sum of the digits is exactly  $n$ .

For example, if the user chooses 13 for  $n$ , the program should print out 49 because  $4 + 9 = 13$ . It would also print 58, 67, and other numbers with the same digit sum. It would not print 48 or 50.

(Suggestion: It might be convenient to write a function called *digitSum*.)

**Answer:**

**Problem 11** Write a complete C++ program that does the following.

1. It asks the user to enter a (single) first name.
2. The program stores the name, but if it is "Freddy", the program changes it to "you".
3. The program says hello to the user, using their name (or changed version).

Here is an example of how the program should work:

```
Who are you?    Max
Hello Max.
```

**Answer:**

**Problem 12** 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:

```
int x;
string s;
```

- (i) Read a user's first name to  $s$  and their age to  $x$ .
- (ii) Print out the number of characters in the string  $s$ .
- (iii) Set  $x$  to be  $1^3 + 2^3 + \dots + 71^3$ , the sum of the cubes of the numbers from 1 to 71.
- (iv) Repeatedly generate and add a random value between 1 and 6 to  $x$ , until the value of  $x$  exceeds 100.
- (v) Read a complete line of text into  $s$  from a user. If their text includes a substring "Queens", print output saying "College", otherwise do nothing.

**Problem 13** Consider the following C++ program.

```

#include <iostream>
using namespace std;

void mystery(int &p, int q) {
    int temp = p;
    p = q;
    q = temp;
}

int main() {
    int p, q;
    for (p = 0; p < 5; p++) cout << p; cout << endl;
    for (q = 0; q < 5; ++q) cout << q;
    cout << endl;
    for (p = 3; p < 6; p++)
        for (q = 1; q <= 3; q++)
            cout << p - q; cout << endl;
    p = 4; q = 14;
    mystery(q, p);
    cout << p << " " << q << endl;
    p = 4; q = 14;
    cout << ++p - q-- << endl;
    return 0;
}

```

What is the output from the program?

**Problem 14** Write header lines (prototypes) for the following functions. Do not attempt to supply the blocks for the functions.

(a) A function called **numberDigits** that is to return the number of digits of an integer.

**Answer:**

(b) A function called **differenceMax** which is to return the difference between the maximum entries in two arrays of integers. (Do not assume that the arrays have the same capacities.)

**Answer:**

(c) A function called **swap** which is used to swap two values of type double.

**Answer:**

(d) A function called **firstCharacter** which is to return the first character in a string.

**Answer:**

(e) A function called **median** which is to return the median (middle valued) entry in an array that holds an odd number of integer entries.

**Answer:**

**Problem 15** Write a function called *plusTax* that uses parameters that specify a price (in cents) and a tax rate (as a percentage). The function calculates the amount of tax, rounded to the nearest cent. (Half cents must round up.) It adds the tax to the price and returns the result.

For example, a program that uses the function follows.

```

int main() {
    int cost = 100;           // cost is 100 cents
    double taxRate = 4.8;    // tax is at 4.8 percent
    cout << "With tax that is " << plusTax(cost, taxRate) << " cents." << endl;
    return 0;
}

```

It should find a tax of 4.8 cents, round up to 5 cents and print:

With tax that is 105 cents.

**Answer:**

**Problem 16** 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,  $n$  that is at most 100.
2. The program reads a value entered by the user. If  $n$  is not positive, or  $n$  is greater than 100, the program should exit.
3. The program reads  $n$  integers from the user and then prints their last digits in reverse order of input.

For example, a run of the program might be as follows:

```
What is n? 7
Enter 7 numbers: 143 259 63 17 12 8 9
9 8 2 7 3 9 3
```

**Answer:**

**Problem 17** Write header lines (prototypes) for the following functions. Do not supply the blocks for the functions.

- (a) A function called **lastDigit** that is used to find the last digit of an integer.

**Answer:**

- (b) A function called **average** which determines the average of 3 integer values.

**Answer:**

- (c) A function called **largest** which is used to find the largest value in an array of double precision data.

**Answer:**

- (d) A function called **print2DArray** which is to print out all of the data in a 2-dimensional array (the array has 100 columns).

**Answer:**

- (e) A function called **sort** which is to sort an array of strings into alphabetical order.

**Answer:**

**Problem 18** Consider the following C++ program.

```

#include <iostream>
using namespace std;

void mystery(int data[], int p, int q) {
    data[p] = data[q];
    data[q] = data[p];
}

void m2(int p, int q) {
    int temp = p;
    q = p;
    p = temp;
}

void print(int data[], int p) {
    for (int i = 0; i < p; i++)
        cout << data[i] << " ";
    cout << endl;
}

main() {
    int scores[8] = {3, 1, 4, 1, 5, 9, 2, 6};
    int quiz[7] = {0, 1, 2, 3, 4, 5, 6};
    print(scores, 3);
    print(quiz, 4);
    mystery(scores, 1, 2);
    print(scores, 5);
    for (int i = 0; i < 3; i++)
        m2(quiz[i], quiz[i+ 1]);
    print(quiz, 6);
}

```

What is the output from the program?

**Problem 19** Write a function called *countChange* that uses four parameters  $q$ ,  $d$ ,  $n$ , and  $p$  and converts the value of  $q$  quarters,  $d$  dimes,  $n$  nickels, and  $p$  cents into dollars.

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

```

int main() {
    int q = 10, d = 5, n = 1, p = 2;
    double x = countChange(q, d, n, p);
    cout << "You have $" << x << endl;
}

```

It should print:

You have \$3.07

**Answer:**

**Problem 20** Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer value,  $r$  that is at most 100.
2. The program reads a value entered by the user. If the value is not in the right range, the program should terminate.
3. The program reads and stores  $r$  integers from the user and then prints a pattern of  $r$  rows of stars, the lengths of which are the other integers entered by the user.

For example, the following represents one run of the program.

```
How many rows? 4
Enter 4 row lengths: 2 7 1 5
**
*****
*
*****
```

**Answer:**

**Problem 21** Write a complete C++ program that first asks a user to do a simple math problem of your choosing. The user enters an answer and the program grades it as right or wrong.

For example the program might ask about  $6 \times 9$  and respond to an incorrect answer of 42 as follows:

```
What is 6 x 9?
42
Wrong!
```

Your program can always ask the same question. **Answer:**

**Problem 22** Write a complete C++ program that asks a user to enter the prices of 100 different grocery items (each price as a decimal showing dollars and cents). The program calculates and prints the total cost of the items.

**Answer:**

**Problem 23**

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,  $x$ .
2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of  $x$  has been entered. (Note positive means greater than 0.)
3. The program prints out  $x$  squares on top of each other, the first with size 1, the second with size 2, and so on.

For example, if the user enters 3 for  $x$  the program should print:

```
*
**
**
***
***
***
```

**Problem 24** Write a function called *percent* that uses two parameters  $x$  and  $y$  and returns the ratio  $x/y$  as a percentage.

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

```
int main() {
    double z;
    z = percent(1.5, 3.0);
    cout << z << endl;
}
```

It should print:

```
50.0
```

because  $1.5/3 = 1/2 = 50\%$ .

**Answer:**

**Problem 25** Write a C++ function called *range* that returns the difference between the largest and smallest elements in an array.

It should be possible to use your function in the following program. (The output from this program is 10 because the difference between the largest element 13 and the smallest element 3 is  $13 - 3 = 10$ ).

```
main() {
    int data[6] = {11, 12, 11, 3, 12, 13};
    int x;
    x = range (data, 6);
    // data is the array to search, 6 is the number of elements of the array
    cout << "The range is: " << x << endl;
}
```

**Answer:**

**Problem 26** Consider the following C++ program.

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

void mystery(int data[], int p, int q) {
    data[p] = data[q];
    data[q] = 0;
}

void print(int data[], int p) {
    for (int i = 0; i < p; i++)
        cout << data[i] << " ";
    cout << endl;
}

main() {
    int scores[8] = {3, 1, 4, 1, 5, 9, 2, 6};
    int quiz[7] = {0, 1, 2, 3, 4, 5, 6};
    print(quiz, 4);
    print(scores, 4);
    mystery(scores, 3, 4);
    print(scores, 8);
    for (int i = 0; i < 3; i++)
        mystery(quiz, i, i+ 1);
    print(quiz, 7);
}
```

What is the output from the program?

**Problem 27** Write C++ functions called *elementSwap* and *swap* that swap either the values of two elements of an array or the values of two variables.

It should be possible to use your function in the following program. (The output from this program is: 4 3 because the values of *x* and *y* are exchanged.)

```
main() {
    int a[6] = {11, 12, 11, 3, 12, 13};
    int x = 3, y = 4;
    elementSwap(a, 0, 5);
    swap(x, y);
    cout << x << " " << y << endl;
}
```

**Answer:**

**Problem 28** Write a complete C++ program that asks a user to enter the 10 quiz scores for each student in a class of 30 students. For each of the 10 quizzes, the program decides which student(s) have got the highest scores and prints their numbers. (Hint: Store quiz data in a table.)

Sample output might look like:

Top Scores:

Quiz 0: Students: 5 17 23

Quiz 1: Students: 2 11 17 26

Quiz 2: Students: 2 17 23 26 27

and so on....

**Answer:**

**Problem 29** Consider the following C++ program. What is the output?

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

main() {
    int i = 1, j = 1, k = 1;
    while (i < 10)
        cout << i++;
    cout << endl;
    while (j < 10)
        cout << ++j;
    cout << endl;
    while (++k < 10)
        cout << k++;
    cout << endl;

    return 0;
}
```

**Problem 30** Write a C++ program that asks a user how many times it should say hello and then says hello the required number of times. For example, a run of the program might produce the following output:

```
How many hellos do you want: 6
Hello Hello Hello Hello Hello Hello
```

**Problem 31** Two numbers are considered as very different if they differ by more than 10. Write a C++ function called `areVeryDifferent` that determines whether two integers are very different.

For example, your function could be used in the following program.

```

int main() {
    int x = 4, y = 10, z = -4;
    if (areVeryDifferent(x, y)) cout << "x and y are very different" << endl;
    if (areVeryDifferent(x, z)) cout << "x and z are very different" << endl;
    if (areVeryDifferent(y, z)) cout << "y and z are very different" << endl;
    return 0;
}

```

The output from this program would be:

```
y and z are very different
```

**Problem 32** Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer value,  $x$  that is at most 100.
2. The program reads a value entered by the user. If the value is not in the right range, the program should terminate.
3. The program reads and stores  $x$  words from the user and then prints them in reverse order.

For example, the following represents one run of the program.

```

How many words? 5
Freddy and Max were absent
absent were Max and Freddy

```

**Answer:**

**Problem 33** Consider the following C++ program.

```

#include <iostream>
using namespace std;

void mystery(int data[], int p, int q) {
    data[p] = data[q] + data[p];
    data[q] = 0;
}

void print(int data[], int p) {
    for (int i = 0; i < p; i++)
        cout << data[i] << " ";
    cout << endl;
}

main() {
    int scores[8] = {3, 1, 4, 1, 5, 9, 2, 6};
    int quiz[7] = {0, 1, 2, 3, 4, 5, 6};
    print(quiz, 7);
    print(scores, 8);
    mystery(scores, 3, 4);
    print(scores, 8);
    for (int i = 1; i < 7; i++)
        mystery(quiz, 0, i);
    print(quiz, 7);
}

```

What is the output from the program?

**Problem 34** Write a complete C++ program that does the following:

1. It generates two random numbers  $x$  and  $y$  each between 1 and 100. (You should use the functions *rand* and *srand*.)
2. It adds  $x$  and  $y$  to make a secret code.
3. It prints the secret code.

For example, if the program generated the numbers  $x = 11$  and  $y = 13$  which add to 24, the output would be:

The secret code is 24.

**Answer:**

**Problem 35** 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,  $x$ .
2. The program reads the value entered by the user.
3. If the value is not positive, the program terminates. Otherwise, the program prints a checkerboard pattern that forms a square of side  $x$ .

For example, if the user enters 5 for  $x$  the program should print the following diagram with 5 lines.

```
* * *
 * *
* * *
 * *
* * *
```

(Hint: How is an even numbered row printed? How about an odd numbered row?)

**Answer:**

**Problem 36** Write a C++ function called *negSum* that returns the sum of all negative elements in an array of integers.

It should be possible to use your function in the following program. (The output from this program is  $-12$  because the negative elements  $-5$ ,  $-4$ , and  $-3$  have a sum of  $-12 = -5 + (-4) + (-3)$ .)

```
main() {
    int data[6] = {-5, -4, 1, 3, 2, -3};
    int x;
    x = negSum (data, 6);
    // data is the array to search, 6 is the number of elements of the array
    cout << "The negative sum is: " << x << endl;
}
```

**Answer:**

**Problem 37** Write header lines (prototypes) for the following functions. Do not supply the blocks for the functions.

- (a) A function called **isOdd** that is used to decide whether an integer is odd.

**Answer:**

- (b) A function called **max** which determines the largest of 3 double precision values.

**Answer:**

- (c) A function called **swap** which is used to swap two integer values.

**Answer:**

(d) A function called **total** which is to find the sum of all entries in an array of integers.

**Answer:**

(e) A function called **maxIndex** which is to find the index of the largest element in an array of double precision values.

**Answer:**

(f) A function called **sort** which is to sort an array of integers into order.

**Answer:**

**Problem 38** Write a complete C++ program that:

1. Asks a user to enter the number of students in a class and the number of quizzes taken by the class.
2. If either of these numbers is less than 1 or more than 99 the program should exit.
3. The program should then prompt the user to enter all of the scores for each of the quizzes, starting with all scores for Quiz 1, followed by all scores for Quiz 2 and so on.
4. The program should print the number of the student with the highest total.

Number students and quizzes starting at 1.

A sample run of the program might look like:

```
How many students: 3
How many quizzes: 4
```

```
Enter scores for Quiz 1: 10 7 0
Enter scores for Quiz 2: 10 10 0
Enter scores for Quiz 3: 10 6 0
Enter scores for Quiz 4: 10 9 0
```

```
Student 1 got the highest total.
```

**Answer:**