

# An Arithmetic Game

## 1 Task A

Write a C++ main function to accomplish the following tasks and name the source code `arithmetic.cpp`.

1. Select an operator randomly from addition ('+'), subtraction ('-'), multiplication ('\*'), division ('/'), and remainder ('%').

Hint: You can create an array of characters (type `char` in C++) with elements '+', '-', '\*', '/', and '%'. Generate a random integer to represent a valid index in this array. Retrieve the operator located at the chosen index in the array.

Warning: To ensure accurate grading in Gradescope, save operators in the order of '+', '-', '\*', '/', and '%'. If the chosen operator differs from the grading script's expectation, you may not receive credits for your work.

2. Generate the first operand as a single-digit integer, that is, an integer in the range [0, 9].
3. If the operator is either division or remainder, generate the second operand as a **non-zero** single-digit integer, that is, an integer in the range [1, 9], otherwise, generate the second operand as a single-digit integer in the range [0, 9].
4. Perform the operation on the operands based on the chosen operator and save the result.
5. Prompt the user with the problem, receive a reply from the user. If the reply is correct, print "true"; otherwise, print "false".

Here is a sample input/output:

```
what is 7 % 6? 1
true
```

And another sample input/output:

```
what is 1 / 9? 0
true
```

Lastly, one more sample input/output:

```
what is 7 % 3? 2
false
```

There are two approaches to generate a random integer in a given range. To ensure compatibility with Gradescope, it is necessary to **use the first approach**. Furthermore, generate random integers in the following order:

1. Choose an integer representing a valid index in the character array with elements '+', '-', '\*', '/', '%'.
2. Then generate the first operand.
3. Finally, generate the second operand.

The first approach involves using the **rand()** from the `<cstdlib>` library along with the remainder operator, as demonstrated in an example using rand function.

```
1 #include <iostream>
2 #include <cstdlib> //rand
3 using namespace std;
4
5 int main() {
6     int v1 = rand() % 100;          // v1 in the range 0 to 99
7     int v2 = rand() % 100 + 1;      // v2 in the range 1 to 100
8     int v3 = rand() % 30 + 1985;    // v3 in the range 1985-2014
9
10    return 0;
11 }
```

In the second approach, utilize the `<random>` library to generate a random integer in the range [1, 6] using combinations of generators and distributions.

```
1 std::default_random_engine generator;
2 std::uniform_int_distribution<int> distribution(1,6);
3 int dice_roll = distribution(generator); // generates number in the range 1..6
```

## 2 Task B

Task B builds upon the code from Task A. Submit the source code as `arithmetic2.cpp`.

1. Enter the number of questions to answer. Ensure that the input is a positive integer. Continue prompting users with “The number of questions cannot be zero or negative. Re-enter: ” and accept input from users until a positive integer is provided.
2. Utilize the code from Task A to generate questions involving addition (+), subtraction (-), multiplication (\*), division (/), and remainder (%), with operands restricted to single-digit integers.
3. Calculate and print the correctness percentage.
4. If the percentage is 90 or higher, print 'A'; if it's 80 or higher, print 'B'; if it's 70 or higher, print 'C'; if it's 60 or higher, print 'D'; otherwise, print 'F'.

Here is a sample run:

```
Enter the total number of questions to ask: -2
The number of questions cannot be zero or negative. Re-enter: -1
The number of questions cannot be zero or negative. Re-enter: 3
```

```
(1) what is 1 / 2? 0
```

```
true
```

```
(2) what is 8 - 0? 8
```

```
true
```

```
(3) what is 5 - 5? 1
```

```
false
```

```
percentage correct: 66.6667%
```

```
letter grade: D
```

Here is another sample run:

```
1 Enter the total number of questions to ask: -1
2 The number of questions cannot be zero or negative. Re-enter: -2
3 The number of questions cannot be zero or negative. Re-enter: -3
4 The number of questions cannot be zero or negative. Re-enter: 5
5
6 (1) what is 4 - 0? 4
7 true
8
9 (2) what is 2 / 3? 0.6667
10 false
11
12 (3) what is 9 * 7? 63
13 true
14
15 (4) what is 7 - 7? 0
16 true
17
18 (5) what is 5 * 4? 20
19 true
20
21 percentage correct: 80%
22 letter grade: B
```

**Warning:** To pass gradescope, the prompts must be exact matches.

The prompt for percentage must be **percentage correct:** followed by a percentage number, ended with %.

The prompt for letter grade must be **letter grade:** .

### 3 Task C

Read a csv (Command-Separated Values) file containing arithmetic expressions and their corresponding answers, generate questions and answer them, calculate and report the correct rate. Based on the correct percentage, print the corresponding letter grade.

For example, consider the following CSV file. Each row contains items separated by commas, where the first item represents an arithmetic expression, and the second item is the corresponding answer for

that expression.

```
1 (3 + 2) % 2,1
2 3 + 2 % 2,3
3 1 + 3 % 2,2
4 1 + 4 % 2,1
5 1 + 4 / 2,3
6 1 / 2 * 3,0
7 1.0 / 2 * 3,1.5
8 1 / 2 * 3.0,0
9 1 / 2.0 * 3,1.5
```

A sample run of the code is as follows.

```
1 (1) what is (3 + 2) % 2? 1
2 true
3
4 (2) what is 3 + 2 % 2? 3
5 true
6
7 (3) what is 1 + 3 % 2? 2
8 true
9
10 (4) what is 1 + 4 % 2? 1
11 true
12
13 (5) what is 1 + 4 / 2? 3
14 true
15
16 (6) what is 1 / 2 * 3? 0
17 true
18
19 (7) what is 1.0 / 2 * 3? 1.5
20 true
21
22 (8) what is 1 / 2 * 3.0? 0
23 true
24
25 (9) what is 1 / 2.0 * 3? 1.5
26 true
27
28 percentage correct: 100%
29 letter grade: A
```

## 4 Task D

Read a csv file containing arithmetic expressions and their corresponding answers, randomly display the questions and answer them, stop if a correct rate of 60% or higher is reached. Report the correct rate.

Here is a sample run:

```

1 (1) what is 1 + 4 / 2? 3
2 true
3
4 (2) what is 3 + 2 % 2? 3
5 true
6
7 (3) what is 1 + 3 % 2? 2
8 true
9
10 (4) what is 1.0 / 2 * 3? 1.5
11 true
12
13 (5) what is (3 + 2) % 2? 1
14 true
15
16 (6) what is 1 / 2.0 * 3? 1.5
17 true
18
19 At least 60% is correct. Stop.

```

Here is another sample run:

```

1 (1) what is 1 + 4 / 2? 1
2 false
3
4 (2) what is 3 + 2 % 2? 0
5 false
6
7 (3) what is 1 + 3 % 2? 2
8 true
9
10 (4) what is 1.0 / 2 * 3? 1.5
11 true
12
13 (5) what is (3 + 2) % 2? 1
14 true
15
16 (6) what is 1 / 2.0 * 3? 1.5
17 true
18
19 (7) what is 1 + 4 % 2? 0
20 false
21
22 (8) what is 1 / 2 * 3? 0
23 true
24
25 (9) what is 1 / 2 * 3.0? 1.5
26 false
27
28 percentage correct: 55.5556%

```

