プログラミング 演習 I　報告書

題目	５．条件分岐命令

実施年月日　　2025 年 05 月 20 日 （ 火 曜日）

提出年月日　　2025 年 05 月 26 日 （ 月 曜日）

実施場所　　制御情報工学科 コンピュータ演習室

学 年	S2	番 号	40	氏 名	八木 睦月

合計	

Numazu National College of Technology
Department of Computer and Control Engineering

1. 問題設定

3 つの整数 a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、a と b は n で割られたときに同じ余りを持つということを、そうでなければそうでないことを表示するプログラムを作成する

2. 問題分析

今回の問題について、以下の点を特に注意することにする

- （概要）端末に入力された数値をもとに条件分岐をして、結果を出力する
- scanf 関数で入力を受け付ける
- scanf 関数の第 1 引数に、任意の書式指定子の文字列リテラルを与える（書式指定子の一覧については後に記載する）
- scanf 関数の第 2 引数には、入力の結果を格納する変数のアドレスを入れる（アドレスの指定には”&”用いる）
- printf 関数で出力する（C 言語には print 関数が存在しない）
- printf 関数の第 1 引数に、任意の書式指定子を含む文字列リテラルを与える
- printf 関数の第 2 引数以降には、書式指定子に代入する変数を順に入力する
- printf 関数および、scanf 関数を使用するため、”stdio.h”を include する
- atoi 関数を使用するため、”stdlib.h”を include する
- スタート関数（main 関数）の戻り値には 0 を指定する（この値が終了コードとなる）
- 文字列の改行には、エスケープ文字の”\n”を用いる
- 余りの計算には、剰余演算子（%）を用いる
- int（整数型）の変数を用いる
- 今回の課題では、プログラミング言語の指定がないため、授業で既習の C 言語以外でも実装を試みる

3. 設計

この問題で作成するプログラムを以下のような流れ図（フローチャート）で示す

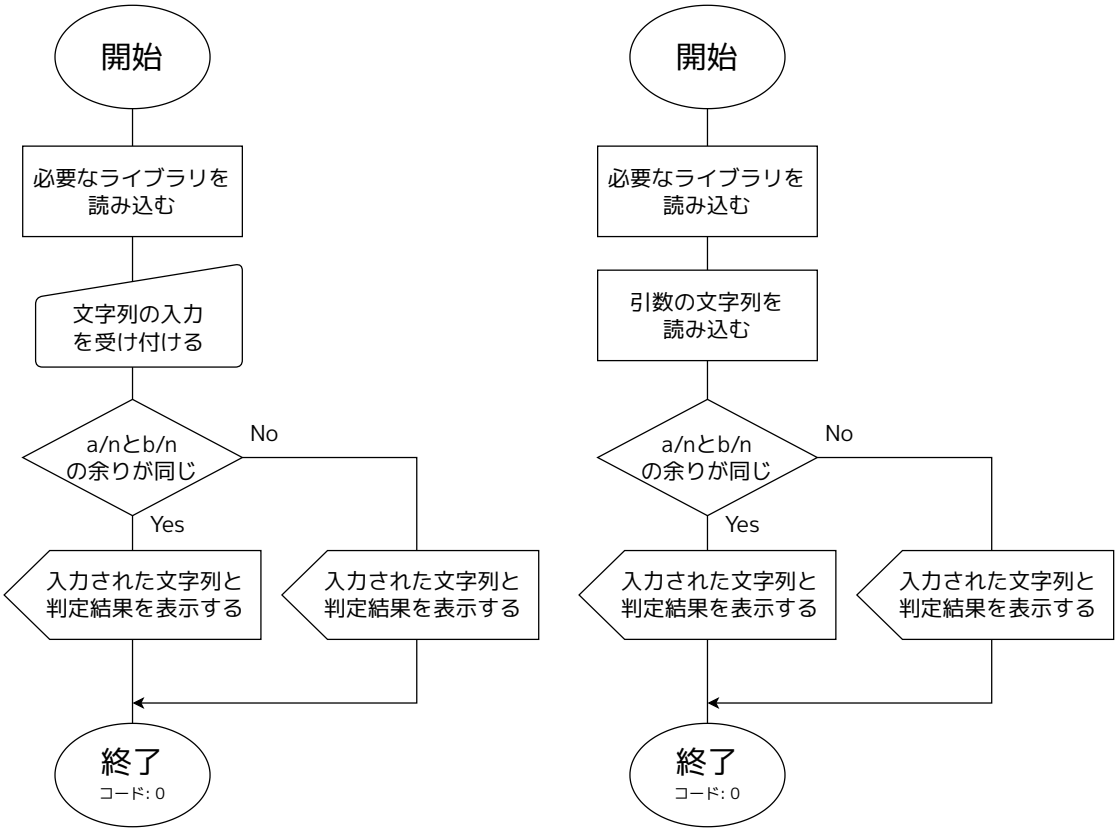

図 1: 左: prog01.c 右: prog02.c（フローチャート）

4. 実装 1

ソースコード 1: prog01.c

```c
/*
3つの整数a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、
a と b は n で割られたときに同じ余りを持つということを、
そうでなければそうでないことを表示するプログラムを作成する
*/

#include <stdio.h>

int main(void) {
    int a, b, n;
    // 整数 a, b, n を入力
    printf(" 被除数a を入力してください: ");
    scanf("%d", &a);
    printf(" 被除数b を入力してください: ");
    scanf("%d", &b);
    printf(" 除数n を入力してください: ");
    scanf("%d", &n);

    // a を n で割った余りと b を n で割った余りが等しいかを判定
    if (a % n == b % n) {
        printf("%d と%d は%d で割られたときに同じ余りを持ちます。\n", a, b, n);
    } else {
        printf("%d と%d は%d で割られたときに異なる余りを持ちます。\n", a, b, n);
    }

    return 0;
}
```

プログラムの解説

7 行目 stdio.h を include する

9 行目 main 関数を定義する

10 行目 変数 a,b,n を整数型として宣言する

12 行目 printf 関数で、文字列を出力する

13 行目 scanf 関数で、整数型の変数 a に値を入力する

14 行目 printf 関数で、文字列を出力する

15 行目 scanf 関数で、整数型の変数 b に値を入力する

16 行目 printf 関数で、文字列を出力する

17 行目 scanf 関数で、整数型の変数 n に値を入力する

20 行目 if 文で、a を n で割った余りと b を n で割った余りが等しいかどうかを判定する

21 行目 真の場合、printf 関数で文字列を出力する

23 行目 偽の場合、printf 関数で文字列を出力する

26 行目 main 関数の戻り値として 0 を返す

用途	説明	変数名	データ型
入力	被除数a	a	int
入力	被除数b	b	int
入力	除数n	n	int

図 2: 変数表（prog01.c）

5. 検証 1

以下は、同じ余りを持つ時の出力である

```
mumu17@GTune-P6I9G60BKABC:~/Documents/プログラミング演習I/5$ ./prog01
 被除数aを入力してください：31
 被除数bを入力してください：25
 除数nを入力してください：3
31と25は3で割られたときに同じ余りを持ちます。
```

図 3: コンソール 1（prog01.c）

以下は、異なる余りを持つ時の出力である

```
mumu17@GTune-P6I9G60BKABC:~/Documents/プログラミング演習I/5$ ./prog01
 被除数aを入力してください：41
 被除数bを入力してください：25
 除数nを入力してください：3
41と25は3で割られたときに異なる余りを持ちます。
```

図 4: コンソール 2（prog01.c）

6. 実装 2

ソースコード 2: prog02.c

```c
/*
   3つの整数a,b,nを入力し、aをnで割った余りとbをnで割った余りが等しければ、
   aとbはnで割られたときに同じ余りを持つということを、
   そうでなければそうでないことを表示するプログラムを作成する
 */

#include <stdio.h>
#include <stdlib.h>

int main(int arg_c, char const *arg_v[]) {
    if(arg_c < 4) {
        printf("3つの引数を入力してください\n 入力例: 被除数a 被除数b 除数n\n");
        return -1;
    }
    // a を n で割った余りと b を n で割った余りが等しいかを判定
    if (atoi(arg_v[1]) % atoi(arg_v[3]) == atoi(arg_v[2]) % atoi(arg_v[3])) {
        printf("%d と%d は%d で割られたときに同じ余りを持ちます。\n", atoi(arg_v[1]), atoi(arg_v[2]), atoi(
            arg_v[3]));
    } else {
        printf("%d と%d は%d で割られたときに異なる余りを持ちます。\n", atoi(arg_v[1]), atoi(arg_v[2]), atoi
            (arg_v[3]));
    }

    return 0;
}
```

プログラムの解説

7 行目 stdio.h を include する

8 行目 stdlib.h を include する

10 行目 main 関数を定義する

11 行目 if 文で、arg_c が 4 未満か判定する

12 行目 真の場合、printf 関数で、文字列を出力する

13 行目 return 文で、-1 を返してプログラムを終了する

16 行目 if 文で、整数型の arg_v[1] を arg_v[3] で割った余りと arg_v[2] を arg_v[3] で割った余りが等しいかどうかを判定する

17 行目 真の場合、printf 関数で文字列を出力する

19 行目 偽の場合、printf 関数で文字列を出力する

22 行目 main 関数の戻り値として 0 を返す

用途	説明	変数名	データ型
入力	引数の個数	arg_c	int
入力	引数の配列	arg_v	char*[]

図 5: 変数表（prog02.c）

7. 検証 2

以下は、同じ余りを持つ時の出力である

```
mumu17@GTune-P6I9G60BKABC:~/Documents/プログラミング演習I/5$ ./prog02 31 25 3
31と 25は 3で割られたときに同じ余りを持ちます。
```

図 6: コンソール 1（prog02.c）

以下は、異なる余りを持つ時の出力である

```
mumu17@GTune-P6I9G60BKABC:~/Documents/プログラミング演習I/5$ ./prog02 41 25 3
41と 25は 3で割られたときに異なる余りを持ちます。
```

図 7: コンソール 2（prog02.c）

以下は、引数の入力に不備がある時の出力である

```
mumu17@GTune-P6I9G60BKABC:~/Documents/プログラミング演習I/5$ ./prog02
3つの引数を入力してください
入力例：被除数a 被除数b 除数n
```

図 8: コンソール 3（prog02.c）

8. 実装 3

ソースコード 3: prog03.py

```
'''
    3つの整数a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、
    a と b は n で割られたときに同じ余りを持つということを、
    そうでなければそうでないことを表示するプログラムを作成する
'''

print("3つの整数を入力してください。\n 入力例：被除数a 被除数b 除数n")
a, b, n = map(int, input().split())

if(n == 0):
    print("n は 0 であってはなりません")
    exit()

if a % n == b % n:
    print("{}と{}は{}で割られたときに同じ余りを持ちます。".format(a, b, n))
else:
    print("{}と{}は{}で割られたときに異なる余りを持ちます。".format(a, b, n))
```

プログラムの解説

7 行目 print 関数で文字列を出力する

8 行目 変数 a,b,n を整数型として宣言する

10 行目 if 文で、n が 0 か判定する

11 行目 真の場合、print 関数で、文字列を出力する

12 行目 exit 文で、プログラムを終了する

14 行目 if 文で、整数型の a を n で割った余りと b を n で割った余りが等しいかどうかを判定する

15 行目 真の場合、print 関数と format 関数で文字列を出力する

17 行目 偽の場合、print 関数と format 関数で文字列を出力する

用途	説明	変数名	データ型
入力	被除数a	a	int
入力	被除数b	b	int
入力	除数n	n	int

図 9: 変数表（prog03.py）

9. 検証 3

以下は、同じ余りを持つ時の出力である

```
C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src>py prog03.py
3つの整数を入力してください。
入力例: 被除数a 被除数b 除数n
31 25 3
31と 25は 3で割られたときに同じ余りを持ちます。
```

図 10: コンソール 1（prog03.py）

以下は、異なる余りを持つ時の出力である

```
C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src>py prog03.py
3つの整数を入力してください。
入力例: 被除数a 被除数b 除数n
41 25 3
41と 25は 3で割られたときに異なる余りを持ちます。
```

図 11: コンソール 2（prog03.py）

以下は、除数 n に 0 が入った時の出力である

```
C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src>py prog03.py
3つの整数を入力してください。
入力例: 被除数a 被除数b 除数n
31 25 0
nは0であってはなりません
```

図 12: コンソール 3（prog03.py）

以下は、引数の入力に不備がある時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src> py prog03.py
3つの整数を入力してください。
入力例: 被除数a 被除数b 除数n

Traceback (most recent call last):
  File "C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src\prog03.py", line
 6, in <module>
    a, b, n = map(int, input().split())
    ^^^^^^^
ValueError: not enough values to unpack (expected 3, got 0)
```

図 13: コンソール 4（prog03.py）

10. 実装 4

ソースコード 4: prog04.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>prog04</title>
</head>
<body>
    <h4 id="url"></h4>
    <input type="text" id="a" placeholder=" 被除数a">
    <input type="text" id="b" placeholder=" 被除数b">
    <input type="text" id="n" placeholder=" 除数n">
    <button id="btn" onclick="javascript:main()">判定</button>
    <h3 id="output"></h3>
    <script src="prog04.js"></script>
</body>
</html>
```

ソースコード 5: prog04.js

```
/*
   3つの整数a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、
   a と b は n で割られたときに同じ余りを持つということを、
   そうでなければそうでないことを表示するプログラムを作成する
 */

var D = document;

D.getElementById("url").textContent = decodeURI(window.location.href);

function main() {
    var a = parseInt(D.getElementById("a").value);
    var b = parseInt(D.getElementById("b").value);
    var n = parseInt(D.getElementById("n").value);

    if (a % n == b % n) {
        document.getElementById("output").innerHTML = a+" と"+b+" は"+n+" で割られたときに
            同じ余りを持ちます。";
    }else {
        document.getElementById("output").innerHTML = a+" と"+b+" は"+n+" で割られたときに
            同じ余りを持ちません。";
    }
}
```

プログラムの解説

7 行目　変数 D に document を代入する

9 行目　id が”url”の要素を取得し、ページの URL 文字列を代入する

11 行目　main 関数を定義する

12 行目　変数 a に、id が”a”の要素の値を整数型で代入する

13 行目　変数 b に、id が”b”の要素の値を整数型で代入する

14 行目　変数 n に、id が”n”の要素の値を整数型で代入する

16 行目　if 文で、a を n で割った余りと b を n で割った余りが等しいかどうかを判定する

17 行目　真の場合、document の id が”output”の要素に、文字列を代入する

19 行目　偽の場合、document の id が”output”の要素に、文字列を代入する

用途	説明	変数名	データ型
演算	document	D	Document
入力	被除数a	a	number
入力	被除数b	b	number
入力	除数n	n	number

図 14: 変数表（prog04.js）

11. 検証 4

以下は、同じ余りを持つ時の出力である

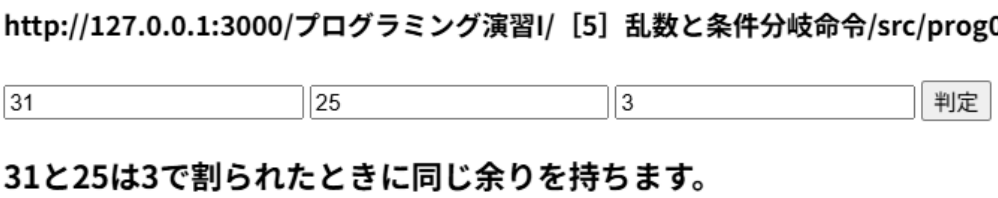

図 15: コンソール 1（prog04.js）

以下は、異なる余りを持つ時の出力である

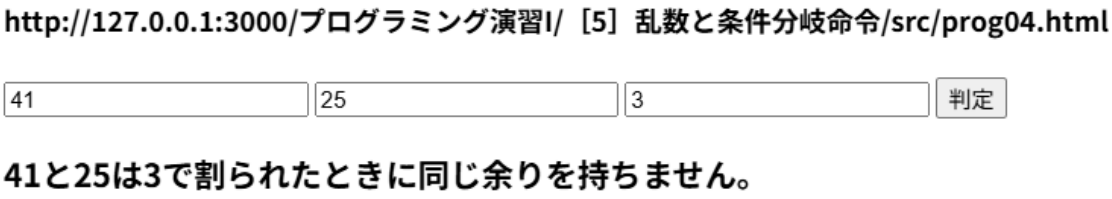

図 16: コンソール 2（prog04.js）

以下は、入力値に不備がある時の出力である

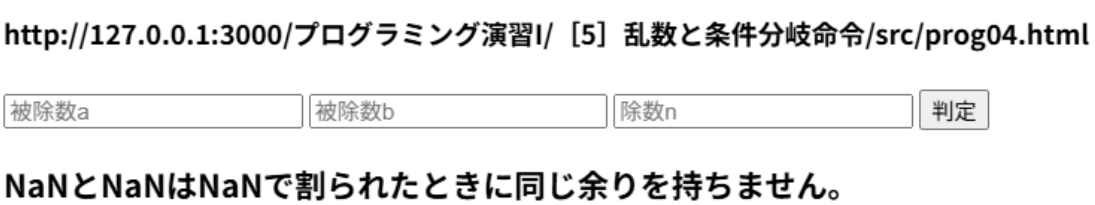

図 17: コンソール 3（prog04.js）

12. 実装 5

ソースコード 6: prog05.java

```
/*
   3つの整数a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、
   a と b は n で割られたときに同じ余りを持つということを、
   そうでなければそうでないことを表示するプログラムを作成する
 */

public class prog05 {
    public static void main(String[] args) {
        if(args.length < 3) {
            System.out.println("3つの引数を入力してください\n 入力例: 被除数a 被除数b 除数n\n");
            return;
        }
        // a を n で割った余りと b を n で割った余りが等しいかを判定
        if (Integer.parseInt(args[0]) % Integer.parseInt(args[2]) == Integer.parseInt(args[1]) % Integer.
            parseInt(args[2])) {
            System.out.printf("%d と%d は%d で割られたときに同じ余りを持ちます。\n", Integer.parseInt(
                args[0]), Integer.parseInt(args[1]), Integer.parseInt(args[2]));
        } else {
            System.out.printf("%d と%d は%d で割られたときに異なる余りを持ちます。\n", Integer.parseInt(
                args[0]), Integer.parseInt(args[1]), Integer.parseInt(args[2]));
        }
        return;
    }
}
```

プログラムの解説

7 行目 prog05 クラスを定義する

8 行目 開始メソッド（main）を定義する

9 行目 if 文で、変数 args の length が 3 未満か判定する

10 行目 真の場合、System.out.println 関数で、文字列を出力する

11 行目 return 文で、プログラムを終了する

14 行目 if 文で、args[0] を args[2] で割った余りと args[1] を args[2] で割った余りが等しいかどうかを判定する

15 行目 真の場合、System.out.printf 関数で文字列を出力する

17 行目 偽の場合、System.out.printf 関数で文字列を出力する

19 行目 main 関数の戻り値を返す

用途	説明	変数名	データ型
入力	引数の配列	args	String[]

図 18: 変数表（prog05.java）

13. 検証 5

以下は、同じ余りを持つ時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］乱数と条件分岐命令\src> java prog05 31 25 3
31と 25は 3で割られたときに同じ余りを持ちます。
```

図 19: コンソール 1（prog05.java）

以下は、異なる余りを持つ時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］乱数と条件分岐命令\src> java prog05 41 25 3
41と 25は 3で割られたときに異なる余りを持ちます。
```

図 20: コンソール 2（prog05.java）

以下は、引数の値に不備がある時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］乱数と条件分岐命令\src> java prog05
3つの引数を入力してください
入力例: 被除数a 被除数b 除数n
```

図 21: コンソール 3（prog05.java）

14. 実装 6

ソースコード 7: prog06.rb

```ruby
# 3つの整数a,b,nを入力し、aをnで割った余りとbをnで割った余りが等しければ、
# aとbはnで割られたときに同じ余りを持つということを、
# そうでなければそうでないことを表示するプログラムを作成する

puts " 被除数aを入力してください"
a = gets.to_i
puts " 被除数bを入力してください"
b = gets.to_i
puts " 被除数nを入力してください"
n = gets.to_i

if a % n == b % n
  puts "#{a}と#{b}は#{n}で割られたときに同じ余りを持ちます。"
else
  puts "#{a}と#{b}は#{n}で割られたときに異なる余りを持ちます。"
end
```

プログラムの解説

5 行目 puts 関数で文字列を出力する

6 行目 変数 a に、gets 関数で入力された値を整数型に変換して代入する

7 行目 puts 関数で文字列を出力する

8 行目 変数 b に、gets 関数で入力された値を整数型に変換して代入する

9 行目 puts 関数で文字列を出力する

10 行目 変数 n に、gets 関数で入力された値を整数型に変換して代入する

12 行目 if 文で、a を n で割った余りと b を n で割った余りが等しいかどうかを判定する

13 行目 真の場合、puts 関数で文字列を出力する

15 行目 偽の場合、puts 関数で文字列を出力する

用途	説明	変数名	データ型
入力	被除数a	a	int
入力	被除数b	b	int
入力	除数n	n	int

図 22: 変数表（prog06.rb）

15. 検証 6

以下は、同じ余りを持つ時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src> ruby prog06.rb
被除数aを入力してください
31
被除数bを入力してください
25
被除数nを入力してください
3
31と 25は 3で割られたときに同じ余りを持ちます。
```

図 23: コンソール 1（prog06.rb）

以下は、異なる余りを持つ時の出力である

```
PS C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src> ruby prog06.rb
被除数aを入力してください
41
被除数bを入力してください
25
被除数nを入力してください
3
41と 25は 3で割られたときに異なる余りを持ちます。
```

図 24: コンソール 2（prog06.rb）

16. 実装 7

ソースコード 8: prog07.bat

```
@echo off
chcp 65001

# 3つの整数a,b,n を入力し、a を n で割った余りと b を n で割った余りが等しければ、
# a と b は n で割られたときに同じ余りを持つということを、
# そうでなければそうでないことを表示するプログラムを作成する

set /p a=被除数a を入力してください:
set /p b=被除数b を入力してください:
set /p n=除数n を入力してください:
set /a "i=a %% n"
set /a "j=b %% n"

if %i%==%j% (
    echo %a%と%b%は%n%で割られたときに同じ余りを持ちます。
) else (
    echo %a%と%b%は%n%で割られたときに異なる余りを持ちます。
)

pause
```

プログラムの解説

1 行目 echo を off に設定する

2 行目 chcp 関数で文字コードを UTF-8 に設定する

8 行目 set 関数で文字列を出力し、変数 a に、入力された値を代入する

9 行目 set 関数で文字列を出力し、変数 b に、入力された値を代入する

10 行目 set 関数で文字列を出力し、変数 n に、入力された値を代入する

11 行目 set 関数で、変数 i に a を n で割った余りを代入する

12 行目 set 関数で、変数 j に b を n で割った余りを代入する

14 行目 if 文で、変数 i と j が等しいかどうかを判定する

13 行目 真の場合、echo 関数で文字列を出力する

15 行目 偽の場合、echo 関数で文字列を出力する

用途	説明	変数名	データ型
入力	被除数a	a	int
入力	被除数b	b	int
入力	除数n	n	int
演算	a/nの余り	i	int
演算	b/nの余り	j	int

図 25: 変数表（prog07.bat）

17. 検証 7

以下は、同じ余りを持つ時の出力である

```
C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src>prog07.bat
Active code page: 65001
被除数aを入力してください:31
被除数bを入力してください:25
除数nを入力してください:3
31と 25は 3で割られたときに同じ余りを持ちます。
```

図 26: コンソール 1（prog07.bat）

以下は、異なる余りを持つ時の出力である

```
C:\Users\moonp\OneDriveNIT\Documents\［7］ Other\TeX\プログラミング演習I\［5］ 乱数と条件分岐命令\src>prog07.bat
Active code page: 65001
被除数aを入力してください:41
被除数bを入力してください:25
除数nを入力してください:3
41と 25は 3で割られたときに異なる余りを持ちます。
```

図 27: コンソール 2（prog07.bat）

18. 補足

以下は、今回のレポートで用いた言語と拡張子の対応表である

言語	拡張子
C	c
Python	py
HTML	html
JavaScript	js
Java	java
Ruby	rb
Batch	bat

図 28: 対応表（言語と拡張子）

考察

整数型の変数に対して、%演算子を使用することで、余りを求めることができる。

剰余演算子（%）は、整数型の変数に対してのみ使用できるため、浮動小数点型の変数に対しては使用できない。

if 文で、条件式を評価する際、比較演算子（==）を使用することで、2 つの値が等しいかどうかを判定できる。

比較演算子の両端でも、普段通り演算子を使用することができる。

C 言語では、printf 関数を使用して出力を行うが、他の言語では print 関数や puts 関数など、言語ごとに異なる出力関数が存在する。

C 言語では、scanf 関数を使用して入力を受け付けるが、他の言語では input 関数や gets 関数など、言語ごとに異なる入力関数が存在する。

C 言語以外の言語でも、文字列の改行には同じくエスケープ文字の”\n”を使用する

println 関数など、言語によっては自動的に改行が行われるものもある。

C 言語では、文字列のフォーマット指定子を使用して、出力の形式を指定することができるが、他の言語では異なった方法で出力形式を指定するものがあった。

C 言語のように、実行前にコンパイルが必要な言語と、実行時にインタプリタが解釈する言語では、プログラムの実行方法が異なる。

C 言語では、main 関数の戻り値として 0 を返すことで、正常終了を示すが、他の言語では異なる方法で終了コードを指定するものがある。

C 言語では、変数の宣言時に型を指定する必要があるが、Python や Ruby などの動的型付け言語では、変数の型を明示的に指定する必要がない。

C 言語では、スタート関数（main 関数）を定義する必要があるが、他の言語ではスタート関数の定義が不要なものもある。

C 言語では、ヘッダーファイルを include することで、標準ライブラリの関数を使用できるが、他の言語では異なる方法でライブラリをインポートするものがある。

それぞれのプログラムに関して、実行後に値を入力するものと、コマンドライン引数で値を渡すものの両方が実装できると考えられる。

引数として値を渡す場合、C 言語や Java では main 関数の引数として受け取ることができるが、Batch などの言語では異なる方法で引数を受け取る。

C 言語や Java では、引数の値を整数型に変換する必要があるが、Python や Ruby などの動的型付け言語では、変換を行わずにそのまま使用できる。

値を整数型に変換する際、C 言語や Java では、atoi 関数や parseInt メソッドの引数に値を入れて変換するが、Ruby では、.to_i メソッドを値の後ろにつけて変換する。

所感

今回の課題は、整数型への変換や、剰余演算子の使用方法、if 文による条件分岐など、C 言語の基本的な文法を学ぶ良い機会となった。

また、異なる言語での同様の処理を実装することで、各言語の特徴や文法の違いを理解することができた。

特に、C 言語以外の言語では、入力や出力の方法が異なるため、それぞれの言語に適した方法を学ぶことができた。

【参考文献】

［1］プログラミング演習 I 配布資料（5 乱数と条件分岐命令）

［2］input() 関数　https://www.python.jp/train/string/input.html　（2025/05/24）

［3］競プロの「list(map(int, input().split()))」って何？

https://kakedashi-engineer.appspot.com/2020/02/22/input-list/　（2025/05/24）

［4］【Java 入門】main メソッドとその引数（args）、戻り値について解説

https://www.sejuku.net/blog/65082　（2025/05/24）

［5］オブジェクト指向スクリプト言語 Ruby リファレンスマニュアル

https://docs.ruby-lang.org/ja/latest/doc/index.html　（2025/05/24）

［6］.bat（バッチファイル）の if コマンド解説。

https://qiita.com/plcherrim/items/8edf3d3d33a0ae86cb5c　（2025/05/24）

［7］コマンドプロンプトの予感　余りを計算する

https://commandprompt.noyokan.com/command/amari.html　（2025/05/24）

［8］UTF-8 のバッチファイルが文字化けする時の対処 3 選

https://nayutari.com/batch-utf8　（2025/05/25）

［9］decodeURI 関数 - JavaScript — MDN

https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/decodeURI（2025/05/25）

【使用ツール】

［1］LaTeX　https://texwiki.texjp.org/LaTeX 入門

［2］draw.io　https://draw.io/