

プログラミング演習I第2回基礎課題 1 レポート

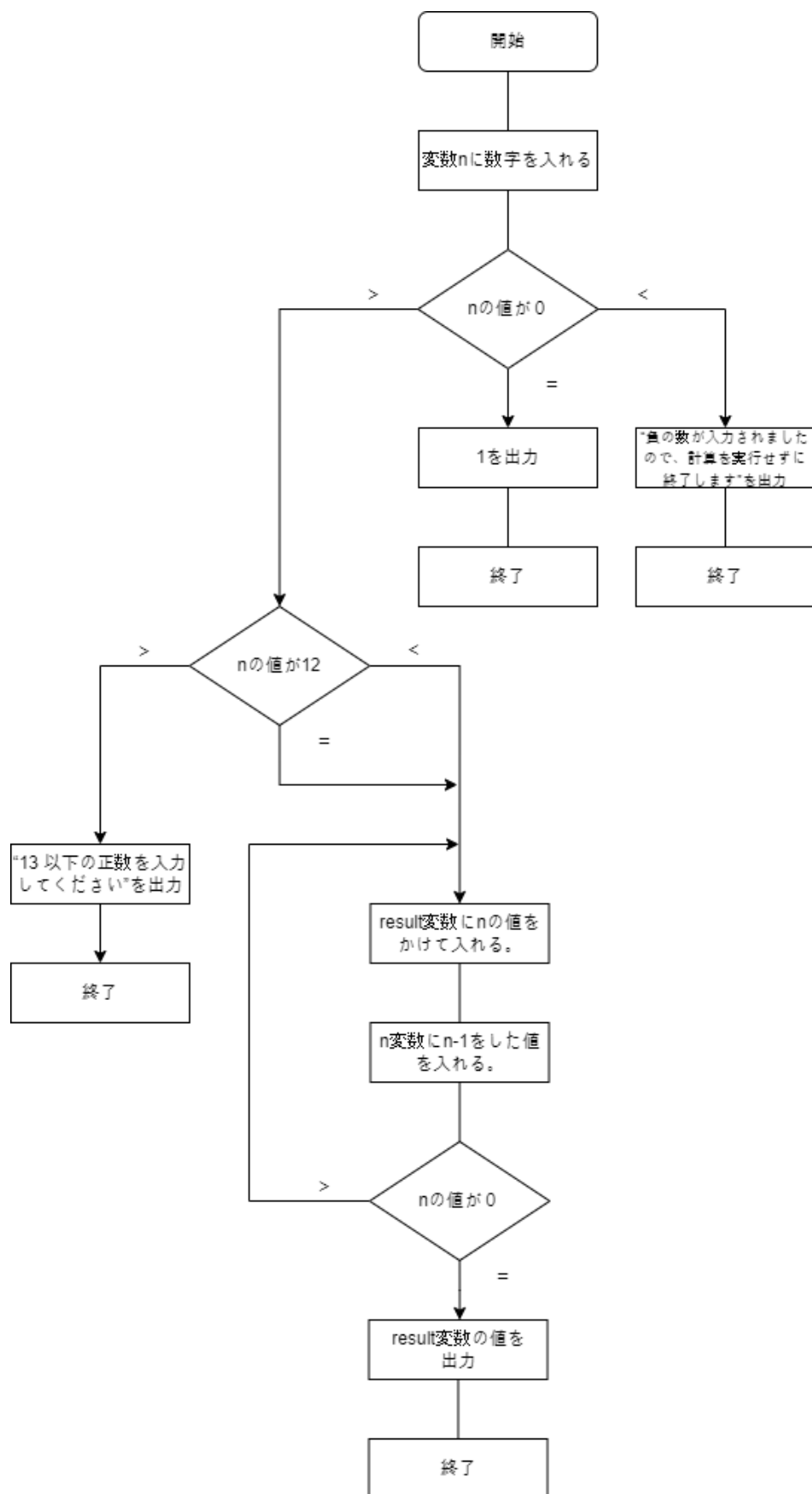
学籍番号：2364902

名前：キム ギュソク

(1) 課題番号と課題内容

- A. 課題番号：基本演習課題 1.
- B. 課題内容： 繰り返し処理で階乗を求める。

(2) フローチャートまたは疑似言語によるアルゴリズムの記述



(3) アルゴリズムが正しいことの説明

A. 入力

- i. ユーザーからの整数を入力してもらうためにscanfを利用して入力された値を変数に入れる。

B. 入力値検証

- i. 入力値が四つの条件により違う出力をするためにif文を利用して三つの条件に分ける。
 1. 条件 I (0 の場合)
 - A. 0の階乗は1であるためそのまま1を出力
 2. 条件 II (負の場合)
 - A. 負の数の階乗は定義されていないため計算せず終了する。
 3. 条件 III (13より大きい場合)
 - A. 結果を正しく計算できないため計算せず終了する。
 4. 条件 IV (1以上12以下の場合)
 - A. 階乗を計算する。

C. 階乗の計算の仕方

i. 階乗の公式

1. $n! = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$

ii. 繰り返し文を利用して計算

1. nの値が1ずつ減りながらかける単純な繰り返し計算なので、繰り返し文(while文)を利用して計算可能。
2. 方法
 - A. nの値が1の時計算が終わるので繰り返し文の条件に“0より大きい”を設定する。
 - B. resultと言う変数(1に初期化した)にnをかけて入れる。
 - C. nの値にn-1を入れる。
 - D. 繰り返し文を脱出するとresult値(階乗計算結果)を出力。

3. 検証

繰り返し1回目： $result = 1 \times n$

繰り返し2回目： $result = 1 \times n \times (n - 1)$

...

繰り返しn-1回目： $1 \times n \times (n - 1) \times \dots \times 2$

繰り返しn回目： $1 \times n \times (n - 1) \times \dots \times 2 \times 1$

結果： $n \times (n - 1) \times \dots \times 2 \times 1$

階乗の公式： $n! = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$

階乗の公式と結果が一致するので正しいと見れる。

(4) ソース・プログラムの説明

A. 実行結果

```
main.c
1  #include <stdio.h>
2
3  int main(void){
4      int n;
5      int result = 1;
6      printf("input number = ");
7      scanf("%d", &n);
8
9      if(n == 0)
10         printf("1");
11     else if(n < 0)
12         printf("負の数が入力されましたので、計算を実行せずに終了します。");
13     else if(n > 12)
14         printf("13 以下の正数を入力してください");
15     else{
16         while(n > 0){
17             result = n * result;
18             n--;
19         }
20         printf("%d", result);
21     }
22 }
```

input

```
input number = 11
39916800
...Program finished with exit code 0
Press ENTER to exit console.
```

B. 説明

```
int main(void){
```

int n; 入力された数字を入れるための変数を宣言。

int result = 1; 階乗計算結果を入れるための初期値 1 を入れた変数

```
printf("input number = ");
```

```
scanf("%d", &n); 入力された数字をnに入れる。
```

```

if(n == 0)    入力された数字が0の場合
    printf("1");    0の階乗は1であるためそのまま1を出力
else if(n < 0)    入力された数字が0より小さい場合
    printf("負の数が入力されましたので、計算を実行せずに
終了します。");
else if(n > 12)    入力された数字が13以上の場合
    printf("13 以下の正数を入力してください");
else{    入力された数字が1以上12以下の場合
    while(n > 0){    nが1の時繰り返し文を終了するための条件
        result = n * result;    階乗の計算
        n--;
    }
    printf("%d", result);
}

```

(5) 考察

- A. まず、ユーザーから入力された数字を条件別に違う行動をするようにif文を利用して条件を四つに分けて満足する数字を入力された場合階乗の計算を行うようにコードを作成した。特に、数字が0の場合階乗の性質で0の階乗は1であるのでそれを注意して条件を作った。また、階乗の計算を具現化するために階乗の公式を把握して同じパターンの繰り返しであることを確認して繰り返し文を利用してコードを作成した。計算の上で計算結果を入れるための変数が必要だと考えresultと言う変数を宣言した。ここで、resultに0を入れると最初の計算で $0 \times n$ となるのでresultには1を入れた。階乗の公式から考え1ずつ減りながら掛け算をするのでnの値を1ずつ減りながらresult変数に入れると階乗の計算ができると思う。この過程をnが1になるまでの繰り返しが必要があるので繰り返す文の条件を0より大きい場合として設定するのが正しいと考えた。入力された数字が13以上の場合正しく計算できないので13以上も正しく計算するための方法を考えることが重要だと思う。

(6) 感想

- A. 階乗の計算を繰り返し文を利用して求める方法についてについて考えながら日常の単純計算をコンピュータにどのように指示するか分かるようになった。また、条件によってif文の使い方が慣れるようになった。