

プログラミング演習I第2回基礎課題 2 レポート

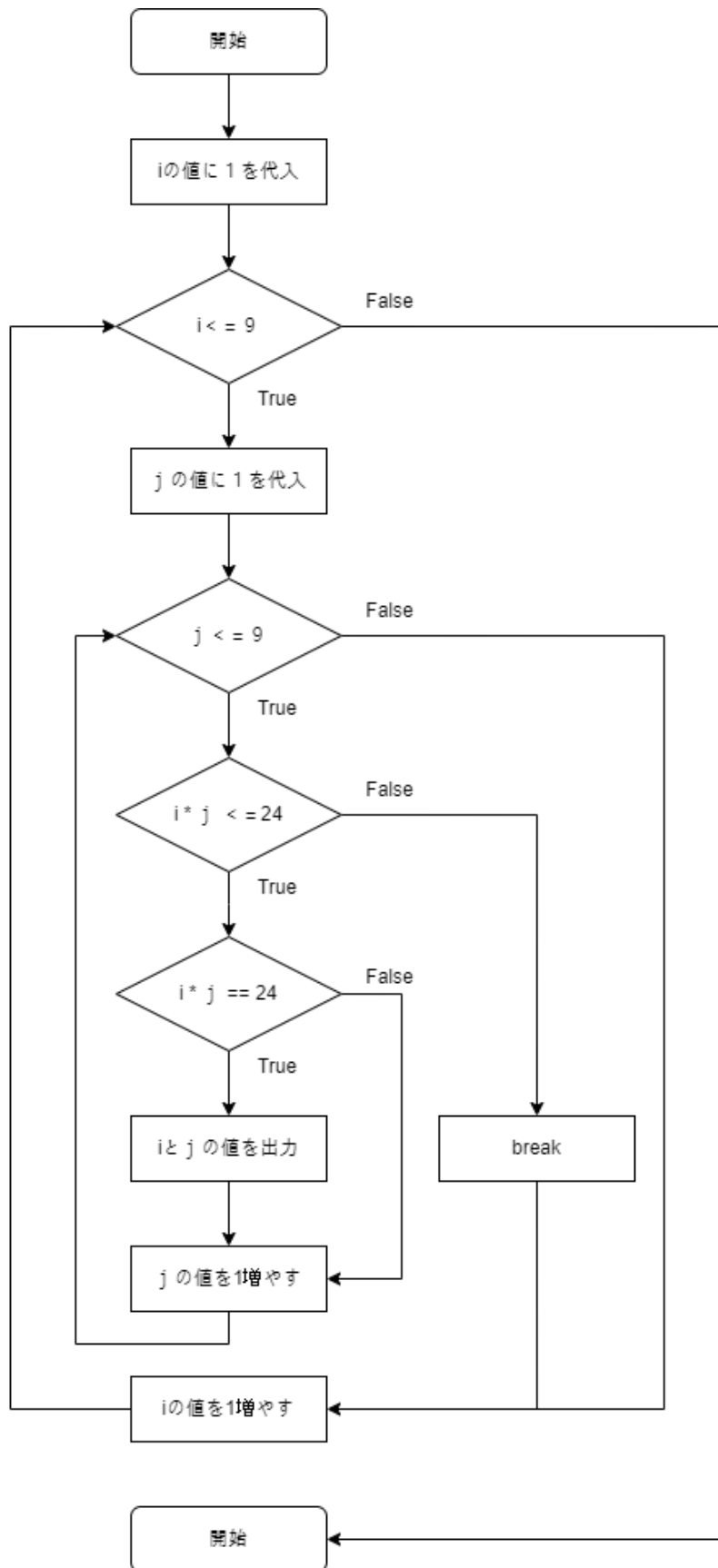
学籍番号：2364902

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 2
- B. 課題内容： 九九で結果が 24 になる数字の組み合わせを表示するプログラムの作成

(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. 正当性

i. 問題解決方法

1. 掛けた値が24であることの確認

A. 条件文(if文)を利用して掛けた値が24である時にprintfを利用して二つの数字を出力するようにすると掛けた値が24であることを確認できる。

2. 掛けた値が24を超えるか確認

A. 1と同じく条件文(if文)を利用して掛けた値が24より大きい場合これ以上の計算は無駄な計算なのでbreakを利用して中のループを脱出することで掛けた値が24を超えるかの確認と無駄な計算を防ぐことができる。

B. 停止性

i. 外側のループ

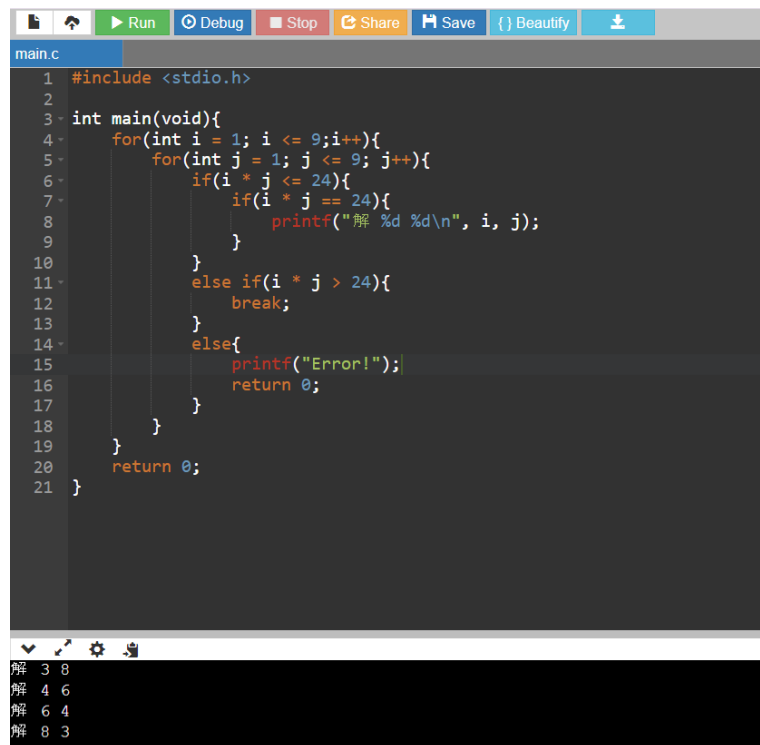
1. 今回は 9×9 の範囲で行うので9回繰り返す必要がある。なので回数のデータである変数iを利用してiが1ずつ増えるにつれてiが9以下である条件を設定することで正しい結果が得られた時に停止するようにコードを作成した。

ii. 内側のループ

1. 外側のループと全く同じく 9×9 の範囲なので9回行うための同じ条件で正しい結果が得られた時に停止するようにした。

(4) ソース・プログラムの説明

A. 実行結果



The screenshot shows a code editor with a file named 'main.c'. The code is a C program that iterates through values of i and j from 1 to 9. It checks if the product i * j is less than or equal to 24. If it is, it prints the result in the format '解 %d %d\n', i, j;'. If the product is greater than 24, it breaks the inner loop. If the product is not equal to 24 and the inner loop ends, it prints 'Error!'. The output window shows the results: '解 3 8', '解 4 6', '解 6 4', and '解 8 3'.

```
1 #include <stdio.h>
2
3 int main(void){
4     for(int i = 1; i <= 9; i++){
5         for(int j = 1; j <= 9; j++){
6             if(i * j <= 24){
7                 if(i * j == 24){
8                     printf("解 %d %d\n", i, j);
9                 }
10            }
11            else if(i * j > 24){
12                break;
13            }
14            else{
15                printf("Error!");
16                return 0;
17            }
18        }
19    }
20    return 0;
21 }
```

解 3 8
解 4 6
解 6 4
解 8 3

B. 説明

```
int main(void){
    for(int i = 1; i <= 9; i++){  $9 \times 9$  を表現するための外側のループ
        for(int j = 1; j <= 9; j++){  $9 \times 9$  を表現するための内側のループ
            if(i * j <= 24){  $24$  より小さいのかを確認するための if 文
                if(i * j == 24){ 掛けた値が  $24$  であることを確認するための if 文
                    printf("解 %d %d\n", i, j); 掛けた値が  $24$  である時の各値
                }
            }
            else if(i * j > 24){  $24$  より大きい場合確認
                break; 無駄な計算を防ぐための break
            }
            else{ 二つの条件を満たさないとき
                printf("Error!"); 条件として有り得ない場合なのでエラーと出力してプログラム終了
            }
        }
    }
    return 0;
}
```

(5) 考察

A. この問題で重要なポイントは 9×9 の計算の仕方、掛けた値が24を超える場合二つである。

i. 9×9 の計算の仕方

		外側のループ (i)								
		1回目	2回目	3回目	4回目	5回目	6回目	7回目	8回目	9回目
内 側 の ル ー プ (j)	1回目	1×1	1×2	1×3	1×4	1×5	1×6	1×7	1×8	1×9
	2回目	2×1	2×2	2×3	2×4	2×5	2×6	2×7	2×8	2×9
	3回目	3×1	3×2	3×3	3×4	3×5	3×6	3×7	3×8	3×9
	4回目	4×1	4×2	4×3	4×4	4×5	4×6	4×7	4×8	4×9
	5回目	5×1	5×2	5×3	5×4	5×5	5×6	5×7	5×8	5×9
	6回目	6×1	6×2	6×3	6×4	6×5	6×6	6×7	6×8	6×9
	7回目	7×1	7×2	7×3	7×4	7×5	7×6	7×7	7×8	7×9
	8回目	8×1	8×2	8×3	8×4	8×5	8×6	8×7	8×8	8×9
	9回目	9×1	9×2	9×3	9×4	9×5	9×6	9×7	9×8	9×9

図1. 9×9 の計算

1. 9×9 の計算をするためには図1のようにループが二つの必要である。外側のループが1回目に入るとその中にある内側のループが1から9まで繰り返して1段の計算を行う。このように外側のループが9まで行くと 9×9 の計算を行うことができる。

ii. 掛けた値が24を超える場合

1. この問題では掛けた値が24になることを見つけることなので掛けた値が24を超えた場合は無駄な計算になる。よって、掛けた値が24を超えた場合ループを脱出することが効率的だと考えられる。また、無駄な計算を取り除くことでプログラムのスピードが上がると考えられるのでこのような制限が必要だと思う。

(6) 感想

A. 9×9 の計算を行うためにどのように反復処理を使えばいいのか分かるいい経験になった。また、24を超えた場合の制限することでプログラムに及ぼす影響を考えることでより効率的なコードを作成することができた。