

提出日：2024/4/18

プログラミング演習 第2回演習レポート

担当教員：杉本 千佳先生

所属：理工学部 数物・電子情報系学科
電子情報システム EP

学年・クラス：2年 Fe1

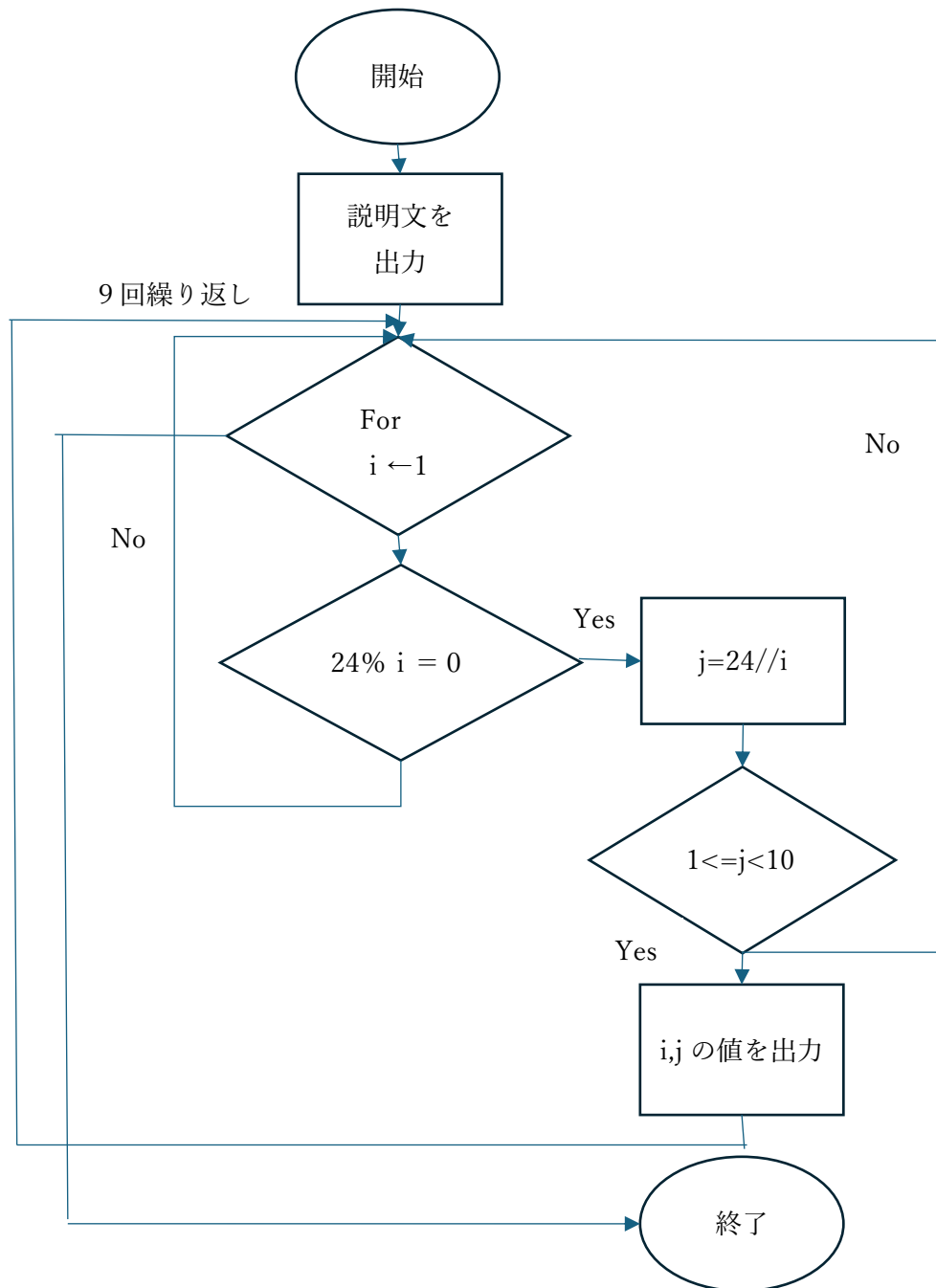
学籍番号：2364092

氏名：熊田 真歩

(1) 課題番号：基本課題2

課題名：九九の掛け算で24になる組み合わせの表示

(2) プログラムのフローチャート

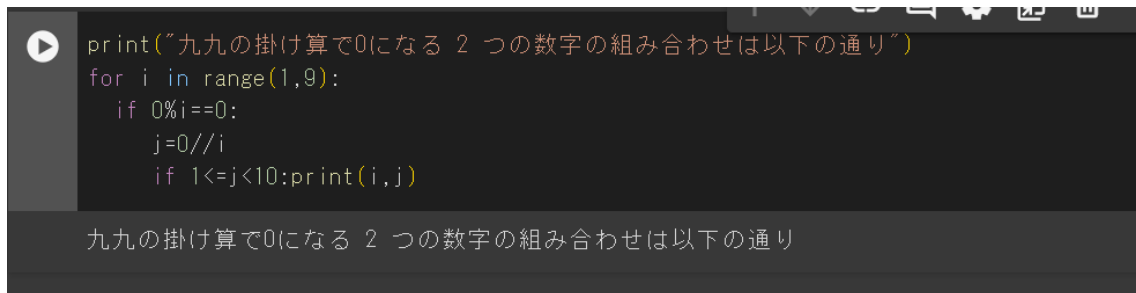


(3) アルゴリズムが「正しいこと」である説明

ここでは 24 以外のすべての整数に対してこのプログラムが停止し、かつ出力結果が正しいか検証しプログラムの正誤性を確かめる。

①値が 0 の時

今回はあくまで九九について考えているので値が 0 の時は値が何も表示されないのが正しい。以下の通りプログラムは終了し値は何も表示されず、正しいと言える。

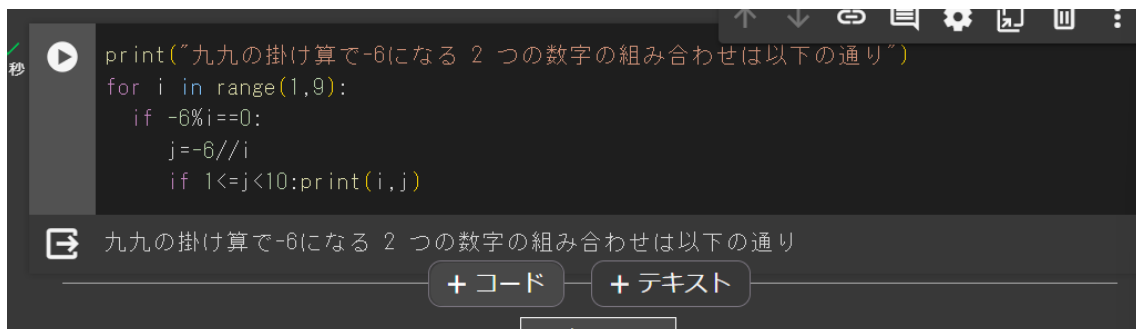


```
print("九九の掛け算で0になる 2 つの数字の組み合わせは以下の通り")
for i in range(1,9):
    if 0%i==0:
        j=0//i
        if 1<=j<10:print(i,j)
```

九九の掛け算で0になる 2 つの数字の組み合わせは以下の通り

②値が負の時

値が負の時においても九九について考えているため値は何も表示されないのが正しい。例として-6 を入れてみると以下の通りプログラムは終了し値は何も出力されず正しいと言える。



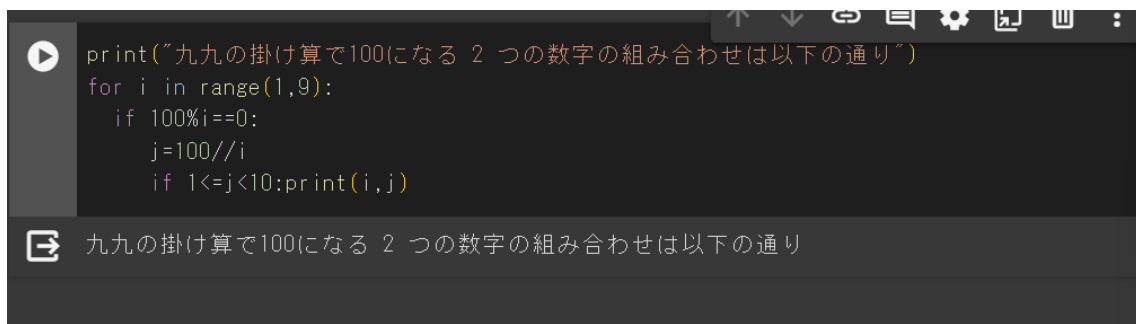
```
print("九九の掛け算で-6になる 2 つの数字の組み合わせは以下の通り")
for i in range(1,9):
    if -6%i==0:
        j=-6//i
        if 1<=j<10:print(i,j)
```

九九の掛け算で-6になる 2 つの数字の組み合わせは以下の通り

+ コード + テキスト

③値が九九の範囲を超えてしまう (100 や 987 など) 時

値が九九の範囲を超えてしまう場合においても値は何も表示されないのが正しい。例として 100 を入れてみると以下の通りプログラムは終了し値は何も出力されず正しいと言える。



```
print("九九の掛け算で100になる 2 つの数字の組み合わせは以下の通り")
for i in range(1,9):
    if 100%i==0:
        j=100//i
        if 1<=j<10:print(i,j)
```

九九の掛け算で100になる 2 つの数字の組み合わせは以下の通り

(4) ソース・プログラムの説明

ここでは、(2) で作成したフローチャートに基づいてソース・プログラムの説明を行う。

1. プログラムの開始
2. 最初に問答無用で"九九の掛け算で 24 になる 2 つの数字の組み合わせは以下の通り"という文を出力する。
3. 九九であるので i を 1 から 9 の範囲で九回繰り返す For 文を作成した。
4. 値が 24 の場合を考えるプログラムであるため、 $24 \div i$ の余りが 0 となる場合に j に $24 \div i$ の商を代入する。($24 \div i$ の余りが 0 とならない場合は For 文に戻り、 i の値を 1 増やし再度プログラムを進める。)
5. 1 以上 10 未満の j のみに対しその時の i 及び j の値を出力する。
6. 再び For 文に戻り 1 値が増加した i に対し再度プログラムを進める。
7. i が 9 の時までのプログラムを終えたらプログラムの終了。

(5) 考察

今回作成したプログラムはある数について、九九の掛け算でその数になる組み合わせについて調べるプログラムを作成した。そのため、九九の表にあるすべての数について結果が正しく表示されるプログラムは作成できている。また、0 や負の数、少数値などの九九の表にないような数を入力すると九九の中に当てはまる数はないため数は出力されないといった観点で作成した。しかし、int 型や float 型といった数以外の例えば char 型などの文字列を代入するとエラーが出てしまう。今回のプログラムはあくまで数を扱うプログラムであるため、アルゴリズムは課題を解くのに適していたと考えたが、char 型の文字列を代入した時には「数を入力してください」などといったメッセージを表示し、再度プログラムが初めから始まるようなつくりになればより正しいプログラムになった。

(7) 感想

フローチャートを初めて作成した。初めてという事もあり、短めではあったがかなり時間がかかった。自分の書いたプログラムの正誤性を判断したり考察をここまで丁寧に行ったことがなかったが、これは非常に重要な作業であると思った。これからもプログラムを書くときは意識していきたい。