提出日：2024/4/26

プログラミング演習　第３回演習レポート

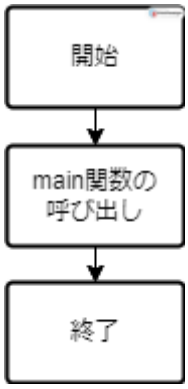
担当教員：杉本　千佳先生
所属：理工学部　数物・電子情報系学科
電子情報システム EP
学年・クラス：２年　Fe1
学籍番号：2364092
氏名：熊田　真歩

（1）課題番号：基本課題２

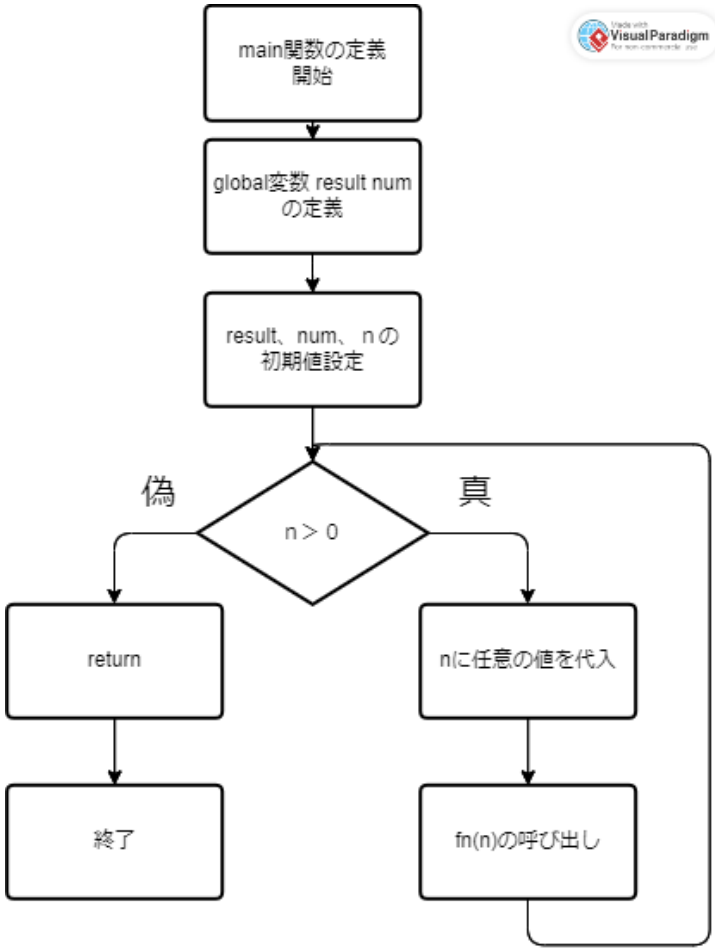
課題名：演習課題１．のプログラム改変

（2）プログラムのフローチャート

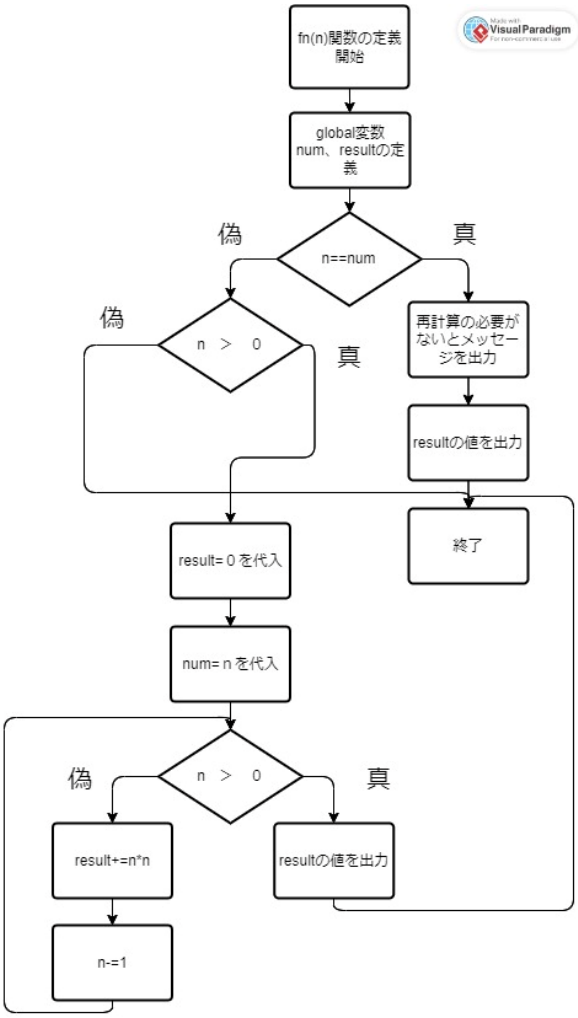
１．プログラム



２．main 関数のフローチャート



３．fn(n)関数のフローチャート



（３）アルゴリズムが「正しいこと」である説明あるいは証明

本プログラムは n にどんな整数値を代入しても常に正当性は保証され、 n の負の整数値を代入した時には常に停止性の保証されるプログラムであるべきである。ここで、n ＝ 0 の時は以下のように出力された。本課題では、前と同じ数を入れた時のみ“再計算する必要はありませんというメッセージの出力を行うというものであったため、これはその指示に反しており、n ＝ 0 の時は厳密にはプログラムの正当性が保証されていないことが分かる。無論、計算結果自体は正しい。

```
main()

n:0
再計算する必要はありません
0
n:-1
```

その他の時は以下のように同じ整数値を二回連続で代入すると二回目は生産する必要がありませんとなり、負の値を代入するとプログラミングが終了し、正しいと言える。

（４）ソース・プログラムの説明

```
def main():                          # main 関数の定義
    global result                    # global 変数 result の定義
    global num                       # global 変数 num の定義
    result = 0                       # int 型変数 result に 0 を代入
    num = 0                          # int 型変数 num に 0 を代入
    n = 1                            # int 型変数 n に 1 を代入
    while n>＝0:                      # n の値が正の時
        n = int(input("n:"))         # n に任意の int 型変数を入力して代入
        fn(n)                        # 関数 fn(n)を呼び出す
    return                           # 何も返さずにプログラムの終了

def fn(n):                           # 関数 fn(n)の定義
    global num                       # global 変数 num の定義
    global result                    # global 変数 result の定義
    if n == num:                     # n の値が num と等しい時
        print('再計算する必要はありません')   # '再計算する必要はありません'というメッセージの出力
        print(result)                # result の値を出力
    elif n > 0:                      # n の値が num と等しいかつ n が正の時
        result = 0                   # int 型変数 result に 0 を代入
        num = n                      # int 型変数 num に n の値を代入
        while n>0:                   #n の値が正の時は繰り返す
            result += n*n            # int 型変数 result の値に n × n の値を加算したものを代入
            n -= 1                   # int 型変数 n に n − 1 の値を代入
        print(result)                # int 型変数 result の値を出力

main()                               # main 関数の呼び出し
```

（5）考察

本プログラムは与えられた自然数の二乗和を計算するというものであった。ここで、プログラムに負の整数を代入した場合は繰り返し構文からはじかれることによりプログラムは終了する。n = int(input("n:"))の指示により代入した n の値は int 型変数であることが指定されている。これは、n に float 型変数である少数値や char 型変数である文字列等を代入するとエラーが起きてしまうことを示す。本プログラムは整数値との指示があったのでこのプログラムは要件を満たしていると言えるが、これを更に汎用性の高いプログラムとするためには n = int(input("n:"))の文を n = input("n:")とし、 n が int 型変数である時は同じくプログラムを継続し、 n が int 型変数でない時はエラーメッセージ等でもう一度 int 型変数値の代入を促すプログラムとすると良いと考えた。

また、本プログラムでは global 変数を導入した。本来、関数内で定義した変数は、それぞれの関数内でのみ有効な local 変数である。しかし、本プログラムでは異なる関数内で global 変数を用いることで同じ変数を用いることでプログラムを簡潔に書くことができた。

更に、n に負の値を代入しない限り永遠に繰り返しが続く本プログラムにおいて、以前計算した値を記憶しておけば同じ値が入力された時に再度計算する必要がなく時間の短縮になる。本プログラムでは記憶しておく値はひとつ前のもののみであったがこのメモリの容量を増やせば一度計算したことのある値はひとつ前でなくても二回目は計算しないというプログラムに改善できる。繰り返し回数が限りなく多いのであれば時間短縮にはなるが、これには本プログラムよりもさらに容量の大きいメモリを要するため、時間か容量かの取捨選択が必要である。

(6) 参考文献、参照情報、謝辞

[1]flowchart を作成するのに用いたサイト

‘Visual Paradigm’ https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=Flowchart&width=11&height=8.5&unit=inch 2024/04/25 アクセス

[2]global 変数について参照したサイト

SAMURAIENGINEER 「【Python 入門】グローバル変数とローカル変数について理解しよう！」 https://www.sejuku.net/blog/58897 2024/4/24 アクセス

（7）感想

今回去年のプログラミング演習の授業であまり理解できていなかった関数の使い方に関する理解が深まった。これは基本課題 1 のプログラムを作成した後に基本課題 2 のプログラムを作成することによりそれらの違いがより明確にわかりやすくなったためである。繰

り返し構文から return を用いることによってプログラムが終了するということも今まで謎に思っていたが、今回の演習を通してしっかりと理解することができた。return による戻り値は完全に理解できているわけではなく試行錯誤しながらプログラムを書かなければいけないのは今の私の課題であると感じた。これからの演習や日々の練習によりこの課題を徐々に克服していきたい。更に、今回は global 変数を用いたがこちらは完全に理解できたと自負している。