プログラミング基礎
第2回

藤江 真也

2021年4月23日

本日の内容

- プログラミング基礎
  - 講義（1時間程度）
    - 資料は13時以降manabaで公開
  - 作業（残り時間）
  - 課題（宿題，manabaに提出）
- 情報処理
  - オンデマンド講義
    - 13時以降manabaで公開
  - 小テスト（manaba）

- アルゴリズム
- フローチャート
- 課題: フローチャート作図

アルゴリズム

足し算

- 問題 1.

15 + 38

- 問題 2.

123 + 284

- 問題 3.

X_1, X_2, Y_1, Y_2 はそれぞれ 0〜9 の数字だとする．それぞれの数字を並べた2ケタの数字 X_2X_1 と Y_2Y_1 を足すときの手順を説明せよ

  - 例: $X_1 = 5, X_2 = 1, Y_1 = 8, Y_2 = 3$ のとき $15 + 38$ を計算する手順となる

できること

※1ケタの数字の足し算ができる
※2ケタの数字に1を足したり，
　2ケタの数字から10を引くことはできる
※足し算の結果が2ケタになるか判断ができる
※2ケタの数字をケタごとに分解することはできる

これはできません！

- $(10X_2 + X_1) + (10Y_2 + Y_1)$ を計算
  - X_2やY_2に10をかけるということはできません
  - 2ケタの数字を直接足すことはできません

できること
※1ケタの数字の足し算ができる
※2ケタの数字に1を足したり，
2ケタの数字から10を引くことはできる
※足し算の結果が2ケタになるか判断ができる
※2ケタの数字をケタごとに分解することはできる

問題 3. 解答例

- X_1とY_1を足したものをZ_1とする
- X_2とY_2を足したものをZ_2とする
- Z_1が10以上の場合，Z_2に1を足し，Z_1から10を引く
- Z_2が10以上の場合，Z_3を1にし，Z_2から10を引く．
  Z_2がもともと10未満の場合はZ_3は0とする
- 以上を行った結果 $Z_3Z_2Z_1$と並べたものが答え

- 問題 4.
  $X_1, X_2, ..., X_i, ..., X_N,$
  $Y_1, Y_2, ..., Y_i, ..., Y_N$
  はそれぞれ 0〜9 の数字とする．
  それぞれの数字を並べたNケタの数字
  $X_NX_{N-1}...X_i...X_2X_1$ と $Y_NY_{N-1}...Y_i...Y_2Y_1$ を足すときの手順を説明せよ

問題4. 解答例

1. X_1 と Y_1 を足したものを Z_1 とする
2. Z_2 を 0 とする
3. i を 1とする
4. Z_i が 10以上の場合，Z_{i+1}に 1 を足し，
   Z_i から10引く
5. i が N なら9.に移る
6. X_{i+1} と Y_{i+1} を足したものを Z_{i+1} に足す
7. i に 1 を足す
8. Z_{i+1} を 0 にして 4. に戻る
9. $Z_{N+1}Z_N ... Z_2 Z_1$ が答えとなる

問題4. 解答例

1. X_1 と Y_1 を足したものを Z_1 とする
2. Z_2 を 0 とする
3. i を 1とする
4. Z_i が 10以上の場合，Z_{i+1}に 1 を足し，Z_i から10引く
5. i が N なら9.に移る
6. X_{i+1} と Y_{i+1} を足したものを Z_{i+1} に足す
7. i に 1 を足す
8. Z_{i+1} を 0 にして 4. に戻る
9. $Z_{N+1}Z_N \ldots Z_2 Z_1$ が答えとなる

足し算・引き算（演算の一種）

問題4. 解答例

1. X_1 と Y_1 を足したものを Z_1 とする
2. Z_2 を 0 とする
3. i を 1とする
4. Z_i が **10以上の場合**，Z_{i+1}に 1 を足し，Z_i から10引く
5. i が **N なら**9.に移る
6. X_{i+1} と Y_{i+1} を足したものを Z_{i+1} に足す
7. i に 1 を足す
8. Z_{i+1} を 0 にして 4. に戻る
9. $Z_{N+1}Z_N \ldots Z_2 Z_1$ が答えとなる

場合分け（条件分岐）

問題4. 解答例

1. X_1 と Y_1 を足したものを Z_1 とする
2. Z_2 を 0 とする
3. i を 1とする
4. Z_i が 10以上の場合，Z_{i+1}に 1 を足し，Z_i から10引く
5. i が N なら9.に移る
6. X_{i+1} と Y_{i+1} を足したものを Z_{i+1} に足す
7. i に 1 を足す
8. Z_{i+1} を 0 にして 4. に戻る
9. $Z_{N+1}Z_N \ldots Z_2 Z_1$ が答えとなる

繰り返し

アルゴリズムに直すとは

- 目標を達成する手順を書き下すこと
  - 目標の例
    - 10 と 20 の和（が求まっている）
    - 「あ」という文字が紙に書かれている

目標は状態

- いくつかの基本処理（演算）の組み合わせで書く必要がある（分解）
  - 基本処理は場合（環境）によって異なる
    - 例）数字の足し算ができる，線が引ける

アルゴリズム

- 複雑な処理を
  - **演算**
  - **条件分岐**
  - **繰り返し**

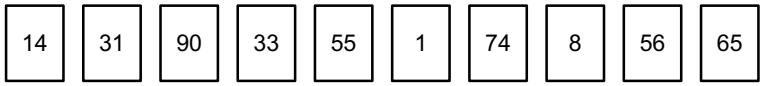
  を組み合わせて手順化したもの
- アルゴリズム化できない処理はプログラムに直せない

もう少し現実的な例

例題 2.1 数列を用いた計算

- 準備
  - 1～100の数字をランダムに10個書き出す
    - 例）14, 31, 90, 33, 55, 1, 74, 8, 56, 65
      http://www.google.co.jp/ で「乱数」と検索すると生成できます
- 準備した数字に対して次の値を計算する方法を考える
  - 最大値，合計値
- ただし，以下の条件を守る
  - 計算用の変数（Xと呼ぶ）を1つだけ使える
  - 同時に見られる数字はXの中身と，10個の数字の中の1つのみ（複数の数値を同時に見ることはできない）

10個の数字

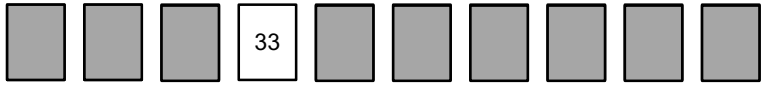


X

ひとつの変数

一度に見られるのは変数Xの中身と，10個の数字のうち1つ

10個の数字



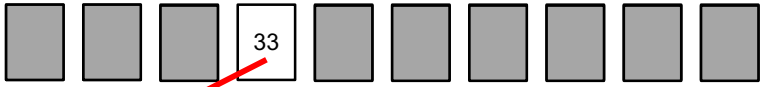
X

ひとつの変数

変数Xの中身は書き換えられる

ただし，今見ている数字か，今見ている数字とXの内容で計算した結果に限る

10個の数字



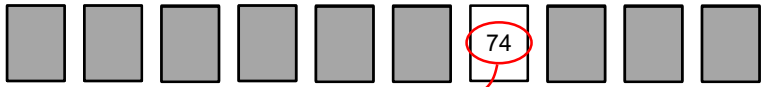
X

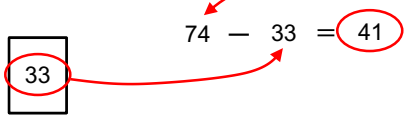
ひとつの変数

変数Xの中身は書き換えられる

ただし，今見ている数字か，今見ている数字とXの内容で計算した結果に限る

10個の数字





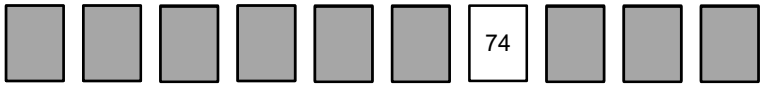
X

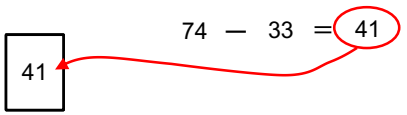
ひとつの変数

変数Xの中身は書き換えられる

ただし，今見ている数字か，今見ている数字とXの中身で計算した結果に限る

10個の数字





X

ひとつの変数

可能な計算

- 四則演算（足し算，引き算，かけ算，わり算）
- 大小比較（どちらの数字の方が大きいか）

アルゴリズム と フローチャート

アルゴリズム

- プログラムは手続きと条件分岐とループの連続

```
#include <stdio.h>

int main(void) {
  int i, sum;

  sum = 0;

  for (i = 0; i < 10; ++i) {
    sum += i;
  }

  printf("%d¥n", i);

  return 0;
}
```

変数sumを0にする

変数iの値を0から9に1加えながらループする

変数sumに変数iを加える

変数sumの内容を画面に出力する

アルゴリズム

- このような一連の手続きのことを
  アルゴリズム(algorithm)と呼ぶ

① 変数sumに0を代入する

② 変数iの値を0から9に1加えながらループする

変数sumに変数iを加える

③ 変数sumの内容を画面に出力する

アルゴリズム

- アルゴリズムは数学の公式に似ているが,
  条件分岐やループがある分より複雑

N個の数値 $x_1, x_2, x_3, \ldots x_N$ の合計値 s を求める

数学

$$s = \sum_{i=1}^{N} x_i$$

アルゴリズム

① s に 0 を代入する

② i に1～N を代入しながらループ

s に x_i を加える

アルゴリズム

- 一連の手続きをアルゴリズム化できれば，プログラム言語に関係なく実装可能

  - 細かい記法は異なるが，手続き・条件分岐・ループがあるのはどの言語も同じ
  - 入出力の仕方などは異なる

逆に…

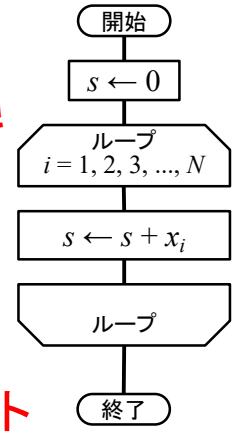
- 一連の手続きをアルゴリズム化できなければ実装は不可能!

アルゴリズムの図による記述法

- アルゴリズムを言葉で書いていると表現が長くなる

① s に 0 を代入する
② i に1～N を代入しながらループ
　s に x_i を加える

図形化！



フローチャート

フローチャート: 開始終了，順次構造

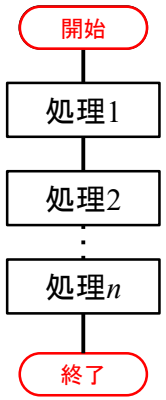
- 端子記号で開始し，端子記号で終了する

端子（terminal）…

- 処理は上から下に実行する順に並べ，線で結ぶ

処理（process）… 処理内容

処理内容は代入や簡単な計算など

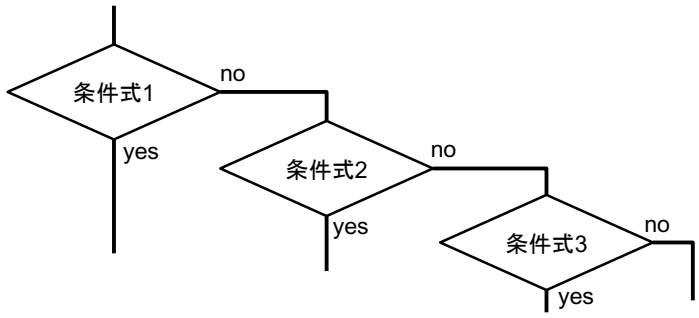


フローチャート: 分岐構造①

- 条件分岐（ifやswitch）には判断記号を使う

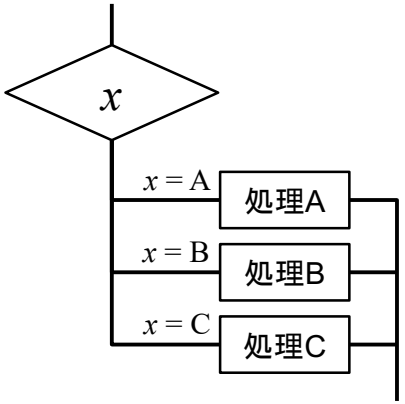
判断（decision）…

- if 文で書ける条件分岐



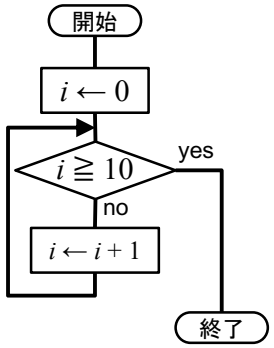
フローチャート: 分岐構造②

- switch文で書ける条件分岐



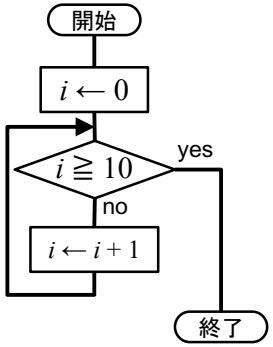
フローチャート: 矢印

- 処理の流れは原則 上から下 または 左から右
- 逆行する場合は必ず矢印をつける
  - 逆行しないときに矢印を付けてもよい

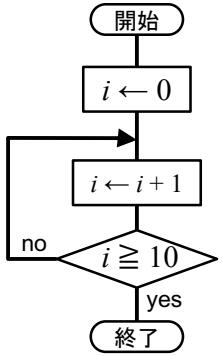


フローチャート: 反復構造①

- 反復（ループ）は，判断記号かループ記号で行う
- while文で書ける構造（前判定型）

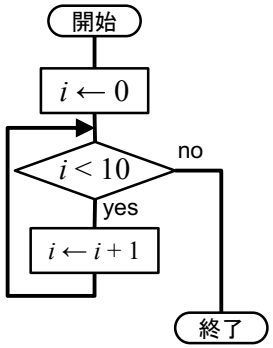


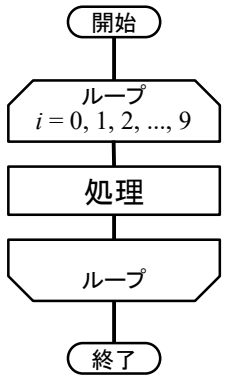
- do-while文で書ける構造（後判定型）



フローチャート: 反復構造②

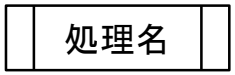
- for文で書ける構造（前判定型）





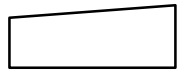
フローチャート: サブルーチン

- 定義済みのまとまった処理(C言語の関数のようなもの)はサブルーチン記号を利用する

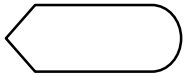
サブルーチン(sub-routine)記号 ... 

フローチャート: 入出力

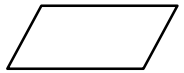
- キーボードからの入力などには手操作入力記号を使う

手操作入力記号 ... 

- 画面への出力などには表示記号を使う

表示記号 ... 

- データの入出力(ファイルなど)にはデータ記号を使う

データ記号 ... 

例題 2.2

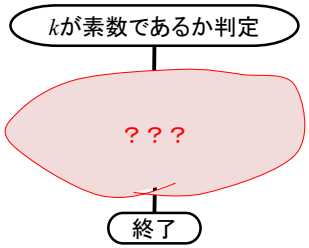
- キーボードから整数値を読み込み，その値が偶数か奇数かを判定し表示するアルゴリズム

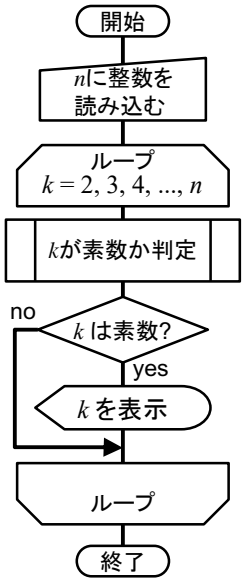
例題 2.3

- キーボードから整数値を読み込み，その値以下の素数を全て表示するアルゴリズム

例題 2.4

- 例題2.3の示した素数を表示するアルゴリズムを，右のように，
  kが素数であるか判定する処理を関数化した場合のフローチャートを描け．
  （下図の???の部分を考える）





課題

※提出方法はmanabaのレポートの欄を確認のこと

※提出期限は4月28日

- 課題 2.1
  - N個の数値 $x_1, x_2, x_3, \ldots x_N$, の平均値 μ を求めて画面に表示するアルゴリズムのフローチャートを作図せよ
- 課題 2.2
  - 整数値を1つキーボードから読み込み，その階乗を求めて，画面に表示するアルゴリズムのフローチャートを作図せよ

プログラミング基礎 第2回
例題解答例

例題2.1 最大値を求める手続き

例）14, 31, 90, 33, 55, 1, 74, 8, 56, 65

- 最初の数字（例だと14）をXに入れる
- 2番目から10番目の数字に対して以下を繰り返す
  - 見ている数字がXに入っている値よりも大きければ，見ている数字をXに入れる
- Xに入っている値が最大値

例題2.1 合計値を求める手続き

例）14, 31, 90, 33, 55, 1, 74, 8, 56, 65

- 最初の数字（例だと14）をXに入れる
- 2番目から10番目の数字に対して以下をくり返す
  - 見ている数字とXに入っている値を足し，その結果をXに入れる
- Xに入っている値が合計値

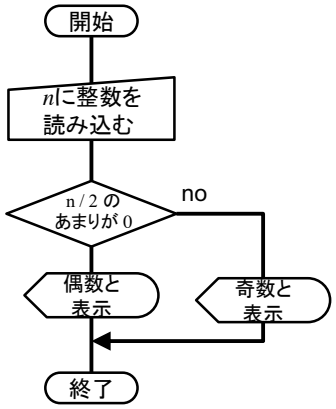
例題2.1 合計値を求める手続き（別解）

例）14, 31, 90, 33, 55, 1, 74, 8, 56, 65

- Xに0を入れる
- 1番目から10番目の数字に対して以下をくり返す
  - 見ている数字とXに入っている値を足し，その結果をXに入れる
- Xに入っている値が合計値

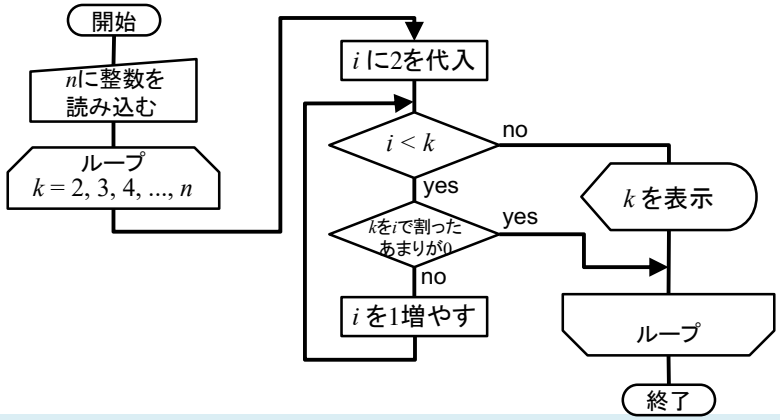
例題 2.2

- キーボードから整数値を読み込み，その値が偶数か奇数かを判定し表示するアルゴリズム



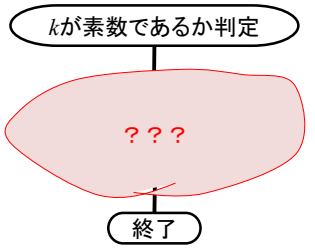
例題 2.3

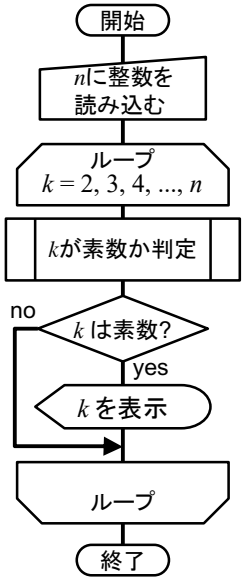
- キーボードから整数値を読み込み，その値以下の素数を全て表示するアルゴリズム



例題 2.4

- 例題2.3の示した素数を表示するアルゴリズムを，右のように，
  kが素数であるか判定する処理を関数化した場合のフローチャートを描け．
  （下図の???の部分を考える）





例題2.4 解答例