## ３. BASICプログラミング演習（基礎）

3.1 BASICとは

BASICは**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode（初心者向け汎用記号命令コード）の略称で，その名の通り初心者でも簡単にプログラムを書くことができるプログラミング言語である。

これからの4週は，先週までに作成したIchigoJam上でBASICのプログラムを実際に動作させながらプログラミングの基礎を学んでいく。

3.2 BASICのプログラムの書き方

BASICのプログラムの例をリスト3.1に示す。

リスト3.1 LEDを点滅させるプログラム

10 LED 0

20 WAIT 30

30 LED 1

40 WAIT 30

50 GOTO 10

60 END

BASICのプログラムは「行番号 命令の種類 命令に必要な値」という行の組み合わせで成り立っている。プログラムは行番号の小さいものから順に実行される。

命令の種類を「関数」とよび，命令に必要な値は「引数（ひきすう）」とよぶ。

IchigoJamで使うことのできる関数一覧は，IchigoJam組み立てキットに同封されている説明書の裏面に記載されている。演習中わからない関数が出てきた時には，説明書を確認すること。

3.3 PCとの接続

シリアル通信ケーブルを用いてPCにIchigoJamを接続する。まず，シリアル通信ケーブルとジャンパーコードを用意し，表3.1のように接続する。さらに、シリアル通信ケーブルのUSBコネクタはPCのUSBポートへ接続する

表3.1 シリアル通信ケーブルとIchigoJamの接続

|  |  |  |
| --- | --- | --- |
| **シリアル通信ケーブル** | **—** | **IchigoJam** |
| TXD （オレンジ） | — | RXD |
| RXD （黄色） | — | TXD |
| GND （黒色） | — | GND |

次に，MicroUSBケーブルをIchigoJamのMicroUSB端子とPCのUSBポートにそれぞれ接続する。そして，IJUtilities\*を起動し，ポートを選択し，IchigoJamとの通信を開始する。ポートは，**COM1以外のもの(COM3など)を選択**する。

IchigoJamのスライドスイッチを左にスライドすると，図3.1のような画面になることを確認する。

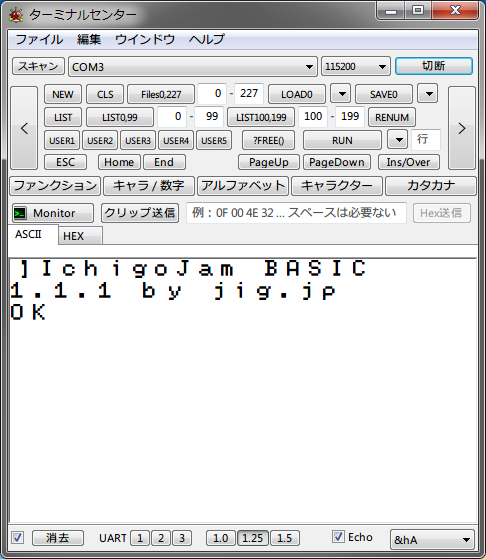


図3.1 IchigoJamを起動した時のIJUtilitiesの表示

\* IJUtilitiesはMicono Utilitiesさんが作成しているツールです．

http://ijutilities.micutil.com/

3.4 IJUtilitiesの使い方

①IJUtilitiesを起動し，IchigoJamの電源を入れる

②「ツール」ウィンドウ（細い画面）の「新規」ボタンを押す

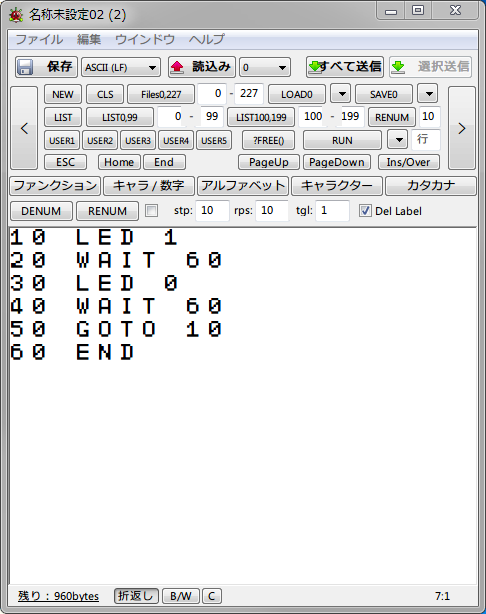
③表示されたウィンドウ（図3.2）にプログラムを打つ

④打ち終わったら「すべて送信」ボタンを押し，「RUN」ボタンを押す（図3-1）

⑤正しく実行できたら，プログラムを入力したウィンドウで，プログラムをコピーし，Wordに貼り付ける（プログラムを入力したウィンドウで右クリック→「すべてを選択」→「コピー」，Wordの画面で右クリック→「貼り付け」）

⑥もし，正しく実行できず，プログラムが終了されなくなった場合，ESCキーを押してプログラムの実行を終了させる。

⑦**NEWボタンを押して**，次のプログラムを作成する。



**NEWボタン**

**すべて送信ボタン**

**RUNボタン**

図3.2 プログラム入力ウィンドウ

以降はBASICを用いたプログラムの演習を示す。演習ごとに演習番号とソースコードをWordに貼り付け，図3.3の書式で報告せよ。なお，提出先やファイル名は以下のようにすること。

提出先（例）:

ReadWrite(des16)(Y:)\総合理工学科\情報システム系\1年実験実習\T1\1-1-40名前

ファイル名(例):

課題1.docx

1年○組○番 名前

PG1-1

10 LED 0

20 WAIT 30

30 LED 1

40 WAIT 40

50 GOTO 10

60 END

[結果] LEDが点滅した

PG1-2

・・・・

図3.3　報告書の書式

PG1-1. LEDの点滅

リスト3.2に示すプログラムを入力し，LEDが点滅することを確認しなさい。

リスト3.2 LEDを点滅させるプログラム

10 LED 0

20 WAIT 60

30 LED 1

40 WAIT 60

50 GOTO 10

60 END

60

PG1-2. 文字の表示1

リスト3.3に示すプログラムを入力し，画面上に文字が表示されることを確認しなさい。

リスト3.3 文字を表示するプログラム

10 PRINT "Hello World!"

20 END

PG1-3. 文字の表示2

リスト3.3のプログラムを参考に，自分の名前（アルファベット）を画面に表示させなさい。

PG1-4. 模様の表示

リスト3.4に示すプログラムを入力し，画面上に模様が表示されることを確認しなさい。

リスト3.4 模様を表示するプログラム

10 PRINT "\*\*\*"

20 PRINT "==="

30 PRINT "+++"

40 END

PG1-5. 四則演算1

リスト3.5に示すプログラムを入力し，計算結果が正しいことを確認しなさい。**A%BはAをBで割った値のあまりを表す。**

リスト3.5 四則演算をするプログラム

10 PRINT 5+3

20 PRINT 10-2

30 PRINT 2\*4

40 PRINT 32/4

50 PRINT 17%9

60 END

PG1-6（応用）. 四則演算2

リスト3.5の計算式の数字を変更して，結果が正しいことを確認しなさい。なお，IchigoJamのBASICでは計算に整数しか用いることができず，割り算の結果が小数になる場合は**小数点以下が切り捨てられる。**

3.5 変数

数学の授業では，数式の中で数字の代わりに*x*や*y*といった文字を用いてきた。プログラムにおいても，「変数」というものが用意されている。変数も数式中の文字と同じように数字を入れておく箱として使うことができる。リスト3.6に示すプログラムは，変数を用いて長方形の面積を計算するプログラムである。

リスト3.6 長方形の面積を計算するプログラム

10 H=4

20 W=5

30 A=H\*W

40 PRINT A

50 END

このプログラムでは，H（HEIGHT; 高さ）という変数に4を，W（WIDTH; 幅）に5を「代入」し，A（AREA; 面積）という変数にそれらを掛けたものを代入している。「代入」とは変数という箱に数字を入れることである。変数をプログラムに用いることで，プログラムが何をしているかわかりやすくし，プログラム中の値を変更しやすくするという効果もある。IchigoJamのBASICでは，**変数名にはアルファベット１文字のみを用いる**ことができ，変数には-32767から32767までの整数を代入することができる。

PG2-1. 変数を用いた四則演算1

次に示すプログラムを入力し，計算結果が正しいかどうかを確認しなさい。

リスト3.7 変数を用いた足し算

10 X=10

20 Y=5

30 Z=X+Y

40 PRINT Z

50 END

PG2-2. 変数を用いた四則演算2

次のプログラムを実行し，計算結果を確認しなさい。

リスト3.8 ３変数の演算

10 X=5

20 Y=4

30 Z=3

40 PRINT X+Y\*Z

50 PRINT (X+Y)\*Z

60 END

PG2-3（応用）. 円の面積と円周を求めるプログラム

半径が6の円の面積と円周を求めるプログラムを作成し，結果が正しいかを確認しなさい。なお，円周率を示す変数P=3とし，以下の形式で出力するプログラムにすること。

AREA: (面積)

LENGTH: (円周の長さ)

PRINT "AREA: "; (変数名)という形で文字列と変数名を;でつなぐことで，上記の出力ができる。

"(ダブルクオーテーション)"で囲まれた文字列は「文字」として認識され，そのまま表示される。逆に""で囲まない文字列は「変数」として認識される。

**参考 –変数の名前の付け方-**

変数に名前をつけるとき，a, b, cなどのように意味のない文字を使うことがよくある。

しかし，プログラムを書くときにはなるべく意味のある文字列を使うほうが良いとされている。プログラムを何人かで書くときや，自分が昔に書いたプログラムを修正するときに，a, b, cだとわかりにくい。そのため，上の例ではWやHなどその変数が示すものの英単語の最初の文字を変数名に用いている。多くのプログラミング言語では1文字だけでなく何文字も使える事が多いため，なるべくわかりやすい変数名をつけるようにしよう（IchigoJamではメモリ容量の関係で１文字しか用いることができない）。

3.6 条件分岐・ループ

条件分岐とは「もしボタンを押されたら音を鳴らす」などのように，条件式が成り立つか成り立たないかで実行するプログラムを分岐する命令のことである。BASICにはIF命令が用意されている。

IF命令を使ったプログラムの例と，その処理の流れ（フローチャート）を次に示す。

リスト3.9 条件分岐のプログラム例1

開始

C=0

LED点灯

LED消灯

Cが5以下

終了

Cに1を足す

True

False

10 C=0

20 LED 1

30 WAIT 30

40 LED 0

50 WAIT 30

60 C=C+1

70 IF C<=5 GOTO 20

80 END

図3.4 条件分岐のフローチャート

このプログラムはLEDの点灯・消灯をするたびに，カウント用の変数Cを1増やしていき，Cが5以下の間，その処理を繰り返す。数学では5以下を*C*≦5と表現するが，BASICでは次の表のように条件を表現する。

表3.2 条件式の記述ルール

|  |  |  |
| --- | --- | --- |
| 条件式 | 意味 | 記述例 |
| 式1 == 式2 | 式1と式2の値が等しい | A==B |
| 式1 != 式2 | 式1と式2の値が等しくない | A!=B |
| 式1 <= 式2 | 式1の値が式2の値以下 | A<=B |
| 式1 >= 式2 | 式1の値が式2の値以上 | A>=B |
| 式1 < 式2 | 式1の値が式2の値より小さい | A<B |
| 式1 > 式2 | 式1の値が式2の値より大きい | A>B |

また，条件式が成り立つ時に複数の式を実行したい時には : で命令をつなぐ。

PG3-1. 数値の判定プログラム

次に示すプログラムを入力し，実行しなさい。

リスト3.11 数値判定プログラム

10 C=1

20 IF C==1 PRINT "#"

30 IF C==2 PRINT "\*"

40 IF C==3 PRINT "="

50 WAIT 30

60 C=C+1

70 IF C<=3 GOTO 20

80 END

PG3-2. ボタンが押されたらLEDを光らせるプログラム（１）

次に示すプログラムを入力し，本体のタクトスイッチを押した時にLEDが点灯するかを確認しなさい。BTN() は，タクトスイッチが押されているなら１，そうでなければ0を表す。

リスト3.11 ボタンが押されたらLEDを光らせるプログラム

10 A=BTN()

20 IF A==1 LED 1

30 IF A==0 LED 0

40 GOTO 10

50 END

PG3-3. ボタンが押されたらLEDを光らせるプログラム（２）

以下のプログラムは，リスト3.11と同様の動作をする。

10 LED BTN()

20 GOTO 10

30 END

3.7 圧電サウンダー

IchigoJamは外部に機器を接続し，制御することも可能である。ここでは，IchigoJam組み立てキットに付属していた「圧電サウンダー」を用いて，音を鳴らす。

圧電サウンダーの接続方法や操作方法は，付属している説明書を参照すること。

PG3-4. 音を鳴らそう1

圧電サウンダーをIchigoJamのGNDとSOUNDにさして，次のプログラムを実行し，音がなることを確認しなさい。

リスト3.12 音を鳴らすプログラム1

10 PLAY "CDEFGABC"

20 END

PG3-5. 音を鳴らそう2

次のプログラムを実行し，音がなることを確認しなさい。

リスト3.13 音を鳴らすプログラム2

10 PLAY "CDE3RCDE3RG4E4D4C4D4E4D4"

20 END

PG3-6(応用). 音を鳴らそう3

次に示す条件を満たすプログラムを作成し，報告しなさい。

1. 「シラソファミレド」という音を鳴らす

2. 1.を5回繰り返す

3. 必ずIF文を用いる

**参考 -IF文とループ処理-**

IF文は「もし○○ならば△△する」という命令である。BASICをはじめとする多くの言語には「もし○○ならば△△する。**そうでなければ□□する。**」ということを示すためのIF ELSE文が用意されている（BASICではIF THEN ELSE文）。例えば「奇数ならば#を表示する。そうでなければ\*を表示する。」といったプログラムなら次のようになる。

10 C=1

20 IF C%2!=0 THEN PRINT "#" ELSE PRINT "\*"

30 C=C+1

40 IF C<10 GOTO 20

50 END

難しそうに見えるかもしれないが，Cを2で割ったあまりが1である数は奇数であり，その時(THEN)には # を表示し，そうでなければ(ELSE) \* を表示している，ということがプログラムを読むことでわかる。

奇数をC%2==1(C%2!=1)や，2n+1で表す手法はよく使うので，覚えておくと良い。

## 4. BASICプログラミング演習（応用）

4.1 入力された数値の受け取り

プログラムを作成するとき，ユーザから数字や文字を受け取ることが多い。BASICでも数字の受け取りが可能である。

リスト4.1 入力された数値を受け取るプログラム

10 INPUT "ANS? ", A

20 PRINT A

30 END

リスト4.1を実行すると，ANS? と表示され，数値を入力すると，その数値がAに代入され，画面に表示される。INPUT()命令は数値のみに用いることができる。

演習PG4-1. 数値を入力して処理を行うプログラム1

以下のプログラムを入力し，正しく実行されることを確認しなさい。

リスト4.2 入力された数値に対して四則演算を行うプログラム

10 INPUT "Number? ", N

20 PRINT "N="; N

30 PRINT "N+2= "; N+2

40 PRINT "N-1= "; N-1

50 PRINT "N\*5= "; N\*5

60 PRINT "N/3= "; N/3

70 PRINT "N%2= "; N%2

80 END

4.2 復習と応用

この節では，後半4週間で学んだことを参考に，自分で課題を解いていく。前半２問は基礎問題，後半２問は応用問題である。

PG4-2. 偶数・奇数を判断するプログラム1

以下の処理の流れで，偶数のときと奇数の時で表示を変えるプログラムを作成しなさい。偶数は2で割ったあまりが0である値，奇数は2で割ったあまりが1である値であることを利用する。

1. 変数Cに１を代入する

2. その変数が偶数か奇数かを判定する

3. 偶数なら$を，奇数なら%を表示する

4. Cの値を１増やす

5. 2〜4の処理を20回繰り返す

PG4-3. 数当てゲーム1

以下の処理の流れで数当てゲームを作成しなさい。

1. 変数Aに，答えの値を入力する

2. 友達に数値を入力してもらう

3. 入力された数値と変数Aを比較する

4. 一致したら「Congratulations!」と表示し，不正解なら2, 3を繰り返す。

PG4-4(応用). 偶数・奇数を判断するプログラム2

以下を参考に，奇数が表示されているときにはLEDを点灯させるプログラムを作成しなさい。処理を一時停止させるためにはWAIT関数を用い，WAIT 60 (60ms処理を停止する)のように使う。

1. 変数Cに1を代入する

2. Cの値を表示する

3. Cの値が偶数ならLEDを消灯し，奇数ならLEDを点灯させる

4. 1秒待つ

5. Cの値を1増やす

6. 1〜5の処理を20回繰り返す

**参考 –WAIT関数-**

コンピュータの処理速度は非常に早く，IchigoJamはクロック周波数が48MHzである。すなわち，1秒間に48\*106回もの処理を行っているということである。この速度でLEDを点滅させると，人間の目には見えないほどの速さで点滅してしまい，正しく動作したかがわからない。そこでWAIT関数を用い，その処理がきちんと行われているのかを確認したり，必要に応じて処理を先延ばししたりする。

PG4-5（応用）. 数当てゲーム2

これまでに習ったことを用いて，PG4-3で作成した数当てゲームを改良しなさい。

例) 正解のときにLEDを光らせる，不正解だった回数を記憶しておいて最後に表示する，

乱数を用いる（下記参考を参照），入力された値が答えより大きいか小さいか表示する 等

**参考 -乱数-**

乱数とは，ランダムな値のことで，プログラミング中にしばしば用いられる。BASICにもRND関数があり，RND(6)（0〜6のランダムな数値を返す）などのように使う。

4.3 [発展] FOR文

この節は，上記の課題がすべて終了した人のための発展課題である。

さて，リスト3.9では，条件が成立したときGOTO命令を用いてループを実現しているが，多くのプログラムではFOR文と呼ばれる構文を用いる。以下にFOR文を用いたプログラムを示す。

リスト4.5 FOR文のプログラム例1

10 FOR C=0 TO 5

20 LED 1

30 WAIT 30

40 LED 0

50 WAIT 30

60 NEXT

70 END

FOR C=0 TO 5はCを0から5まで1ずつ増やしていく命令である。

上記のプログラムでは20行目から50行目まで命令を実行した後，60行目のNEXT文で10行目に戻り，Cに1を加算する。そのような処理を実行し，C=5になるとループを抜け，70行目によりプログラムの処理が終了する。

FOR NEXT文は更に細かくプログラムを修正することができ，NEXT文の実行時にCに加算する値を変更することができる。要するに2刻みで値を加算したり，5刻みで値を加算したりできる。リスト4.6にサンプルプログラムを示す。

リスト4.6 FOR文のプログラム例2

10 FOR C=1 TO 10 STEP 2

20 PRINT C

30 NEXT

40 END

リスト4.6を実行すると，2刻みで値が表示される。具体的には，「1, 3, 5, 7, 9」となる。

以下では，FOR NEXT文を利用した課題に挑戦してみよう。

PG4-5（応用）. FOR文を使ってみよう1

リスト4.5に示したサンプルプログラムを実行し，LEDが5度点滅することを確認せよ。

PG4-6（応用）. FOR文を使ってみよう2

リスト4.6に示したサンプルプログラムを実行し，表示される内容を記録せよ。

PG4-7（応用）. 数値判定プログラム

PG-10 リスト3.11に示したプログラムをFOR文で書き直し，実行せよ。

PG4-8（応用）. 偶奇判定プログラム

PG-11「参考 -IF文とループ処理-」中で示したプログラムをFOR文で書き直し，実行せよ。

PG4-9（応用）. 九九を計算させよう

FOR文を2つ入れ子にし，九九を計算するプログラムを作成せよ。なお以下のような出力にすること。

1\*1=1

1\*2=2

（中略）

9\*9=81

なお，リスト4.7を実行すると，00から99までの数値を表示することができる。

リスト4.7 FOR文のプログラム例3

10 FOR A=0 TO 9

20 FOR B=0 TO 9

30 PRINT A; B

40 NEXT

50 NEXT

60 END

2016/03/17　R. Hagihara　作成

2016/03/31　N.　Yanuki　編集

2016/05/19　N.　Yanuki　修正

2017/XX/XX R. Hagihara 修正