誰でもわかるプログラム講座

高校３年２組　織谷康之

０．　はじめに

プログラムと聞くと、「えー、難しいんじゃないの？」というイメージがあるかもしれませんが、実際はそんなことはないんですよ。これを読み終わったときには、あなたもちょっとしたものなら作れるようになるはずです！

１．　プログラムの言語

ここでは、“BASIC”という言語を扱いたいと思います。そう、教科書の一番後ろに載ってて授業では見なかったことにされがちな、あれです。“BASIC”とは英語で“基本の”という意味ですが、プログラムの“BASIC”とは、“Beginner's All-purpose Symbolic Instruction Code”の頭文字なんです。そして、BASICの最大の特徴は、初心者でも簡単に理解できる（限りなく英語に近い）ことなんですよね。さて、BASICの紹介はこれくらいにして、本題に入りましょう。

１－１　変数、文字列の記号

まず、変数と文字列の形式から話を始めましょう。変数の名前は半角英数で後に述べるような命令とかぶらなければなんでもよいんです。変数全部が全部、ＡとかＢとかだと、長いプログラムを作るときはどれがどれか分からなくなってしまうので、たとえば、時間の変数はそのまま“jikan”などといったもの（“time”はtime$という命令があるから駄目）がよいでしょう。次に、文字列ですが、この名前の付け方も先の変数の時と全く同じです。ただ、名前の最後に“$”をつけてください。“namae$”という風にです。

さらに、変数名や文字列名のあとに(n)とつけると、添え字の扱いになります。この時、ｎというのは０以上の数や、変数を入れます。ただし、これには少し制約があるので、それについては後の命令の項で述べます。

１－２　四則演算

次に、基本的な演算です。

A＋B　A－B　A＊B　A／B：加減乗除
A＞B　A＜B　　：大なり　小なり
A＞=B　A＜＝B　A＜＞B　：大なりイコール　小なりイコール　ノットイコール

とまあ、この辺でしょう。もちろん、（　）も使うことは可能です。

１－３　命令

さて、つぎはプログラムの命令の文法です。英語で言うところの語法といったところでしょうか。ここで挙げるのはほんの基本的なものですが、これらを組み合わせれば、アイディア次第でほとんどのプログラムは組むことができるんです！（事実、文化祭の展示プログラムもこれ以外のものは、ほとんどグラフィック関係しか使われていません）

① print　A　/　print “文字”　/　print　A$
その後の数や文字列を表示します。このとき、最後に；（セミコロン）をつけると、改行せずにそのまま次の print 文を実行します。

② if　{条件}　then　{命令}
条件を満たすときにその後の命令を実行します。

③ for　A＝{始まりの数}　to　{終わりの数}　/　next　A
Aに始まりの数を代入し、そのまま次の操作を行います。その後、後にある next　A のところに来たら、Aの数を１増やし、for の行に戻ってそのままさっきやった操作を繰り返します。それを、Aが終わりの数と等しくなるまで繰り返します。

④ input　A　/　input　A$
？の後に入力た数や文字列をAやA＄に代入します。

⑤ goto {行番号}
行番号の行に飛びます。

⑥ gosub {行番号}　/　return
行番号の行に飛び、return で戻ってきます。

⑦ dim A(最大の数)
前で話した添え字の最大の数を定義します。

⑧ int (数) / int (A)
数や変数の小数点以下を切り捨てます。

⑨ randomize val(right$(time$,2)) / A=rnd
randomize で乱数表を定義し、rnd で０以上１未満のある数をランダムで出力します。int (rnd＊n) で０以上ｎ未満のある数をランダムで出力します。

⑩ sqr (数) / sqr (A)
数や変数の平方根を出力します。

⑪ end
プログラムの最後に入力し、プログラムを終了させます。

２． プログラムを組む

では、実際にプログラムを組んでみましょう。ここではオーソドックスに、“ある入力したｎの数以下の素数をすべて表示する”というプログラムを作成してみましょう。また、組むにあたって、行の頭に行番号をふることを忘れないようにするとともに、できるだけパソコンにさせる計算の量を少なくして、処理の速いプログラムを作ることを心がけましょう。

Program Ⅰ

```
10   print "いくつまで？"
20   input n
30   for A=2 to n
40   ama=1
50   for B=2 to A-1
60   if A-B＊int(A/B)=0 then ama=0
70   next B
80   if ama=1 then print A;
90   next A
100   end
```

命令の前の数字は行番号を表します。この番号で行を区別します。まず、20 行目で最大の数を聞き、素数の“その数以下の整数で割り切れない”という性質を利

用して、60 行目で実際に n－１以下の数で割って、割り切れなければ ama は１のままだし、割り切れれば ama は０となり、ama が１であれば、最後にその数を表示しています。

しかし、このプログラム、まだ改良の余地があるのはわかりますか？上のものよりも、次のもののほうが、n が大きくなったとき速度が速くなります。

Program Ⅱ

```
10   print “いくつまで？”
20   input n
30   for A=2 to n
40   ama=1
50   for B=2 to int(sqr(A))
60   if A-B＊int(A/B)=0 then ama=0
70   next B
80   if ama=1 then print A;
90   next A
100   end
```

上のように、二重下線部を変えてみました。これは、“その数以下の整数で割り切れない”ことは、“その数の平方根以下の整数で割り切れない”ことと同値であるからです。

また、まったく違うアプローチで、次のようなものも作ることができます。

Program Ⅲ

```
10   print “いくつまで？”
20   input n
30   dim p(n)
40   for a=2 to n
50   p(a)=0
60   next   a
70   for a=2 to n-1
80   if p(a)=0 then print a;:for b=2 to int(n/a):p(b)=1:next b
90   next a
100   end
```

それぞれのプログラムを実際に走らせて、その処理にかかった時間を比較してみましょう。ｎには1000を代入します。

Program Ⅰ	３分２１秒
Program Ⅱ	３分１９秒
Program Ⅲ	２秒

いまいち、ⅠとⅡの差が出ませんでしたが、Ⅲのスピードは格別ですね。ちなみに、ｎ＝10000としてもⅢは１４秒しかかかりませんでした。これが、はじめに言った、プログラムの効率なのです。

３．応用編

さて、次に、数Ｂの教科書にもある。あみだくじプログラムをちょっとアレンジを加えながら作ってみましょう。流れとしては、

正解の場所を選ぶ→ランダムで棒を配置→並び替え→選択を入力→正解発表

としてみると、

```
10  randomize val(right$(time$,2))
20  dim bou(4)
30  seikai=int(rnd＊4)+1
40  print “何番を選ぶ？（１～４）”
50  input kai
60  print “１　２　３　４”
70  for a=1 to 4
80  bou(a)=int(rnd＊4)
90  if bou(a)=1 then print “｜－｜　　｜　　｜”
100  if bou(a)=2 then print “｜　　｜－｜　　｜”
110  if bou(a)=3 then print “｜　　｜　　｜－｜”
120  next a
130  kotae=seikai
140  for a=1 to 4
150  if bou(a)=1 and kotae=1 then kotae=2:goto 210
```

```
160   if bou(a)=1 and kotae=2 then kotae=1:goto 210
170   if bou(a)=2 and kotae=2 then kotae=3:goto 210
180   if bou(a)=2 and kotae=3 then kotae=2:goto 210
190   if bou(a)=3 and kotae=3 then kotae=4:goto 210
200   if bou(a)=3 and kotae=4 then kotae=3:goto 210
210   next a
220   for a=1 to 4
230   if a=kotae then print "○   ";
240   if a><kotae then print "×   ";
250   next a
260   print ""
270   if kai=seikai then print "当たり"
280   if kai><seikai then print "はずれ"
290   end
```

といった感じになります。このように、どんなプログラムも、小さな命令の集まりによって形成されているのです。

４．最後に

初めてこの部誌というものを書かせてもらって、こんなにも大変なものだとは。私の書いたこの稚拙な文章で、一人でもプログラミングに興味を持って頂けたら幸いです。

このBASICというものはPC98などの旧式のパソコンによく見られていたんですが、時代の流れとともに衰退していきました。しかし、そこで培われた財産を無くすまいと、教科書にも掲載され、センター試験にも出題されているのです。

また、“Active Basic”や、“N88互換BASIC for Windows 95”などを使えば、多少の語法の変更はあるものの、作成したプログラムをWindowsのパソコンでも走らせることができるんです。

この文章を最後まで読んでくださって、ありがとうございました。

fin