愛知県立三谷水産高等学校　教諭　山田　学

海洋情報技術〔プログラミング〕

指導書

ソフトウェアの準備

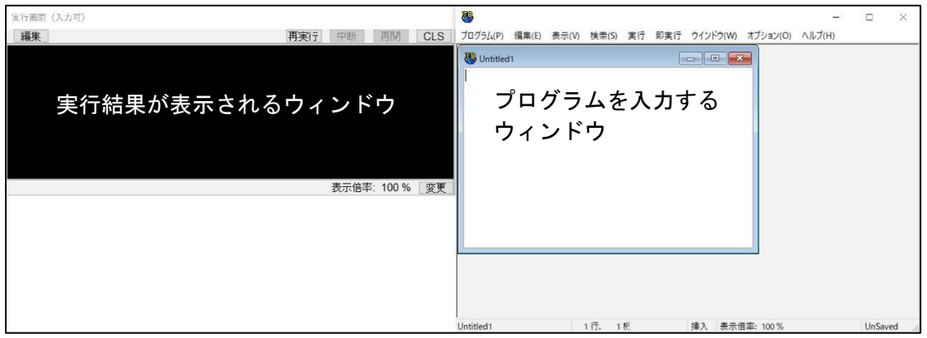
TinyBasic ソフトウェアのダウンロード先

https://tbasic.org/

ファイルをダウンロードし、展開したら、実行ファイル TBasic を生徒の実習用フォルダや USB メモリにコピーして使用する。

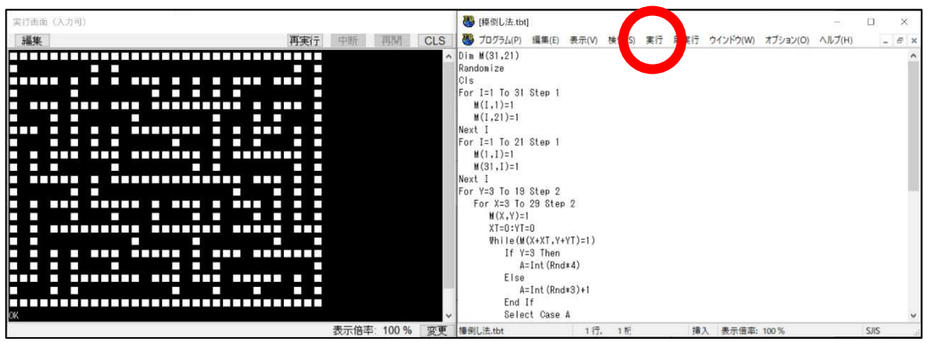
ソフトウェアの起動

TBasic をダブルクリックすると、ソフトウェアが起動する。



プログラムの入力と実行

プログラム入力ウィンドウにプログラムを入力して、
［実行］メニューから［プログラムの実行］を選ぶ、または［Ｆ９］キーを押すことで、
プログラムが実行される。



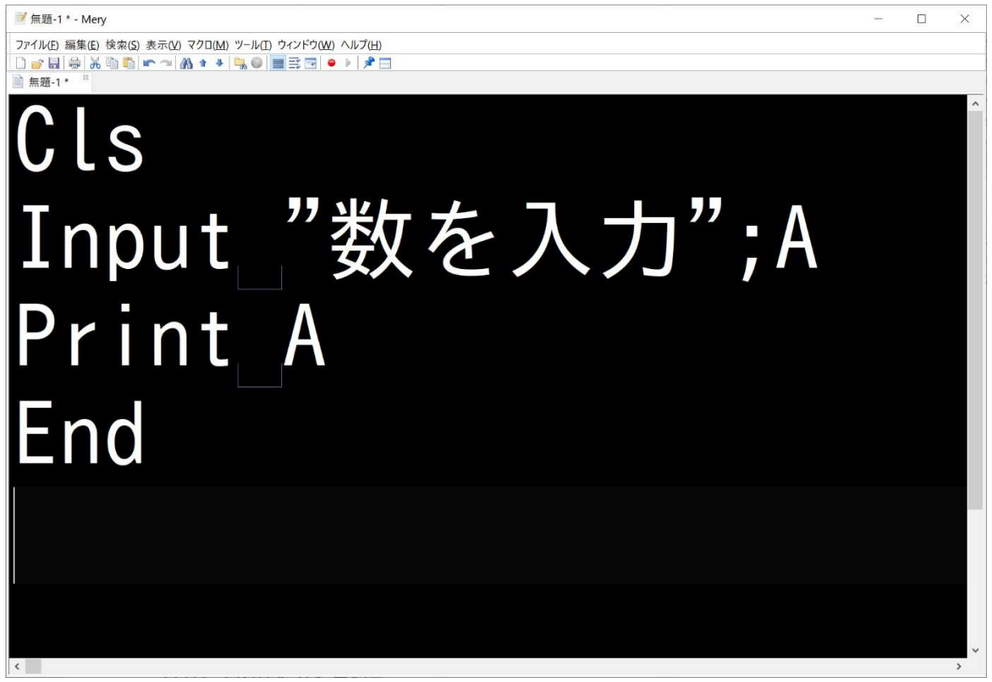
フォントのインストール

プログラムは通常フォントでは、区別がつきにくい文字があるため、プログラミング用のフォントをインストールして使用すると便利です。

Century	MS ゴシック	Takao ゴシック	Ricty(ﾘｸﾃｨ)
0O　1Il	0O　1Il	0O　1Il	0O　1Il

プログラムの投影

Windows のメモ帳 やエディター を使用して、フォントサイズを大きくし、プロジェクターに投影すると便利です。



添付のエディターは [F11]キーを押すと全画面表示に切り替わります。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　１時間目

Step01　文字を表示するプログラム

```
Cls
Print ”HELLO”
```

※プログラムは、上から１つずつ実行されていきます。
　１行目の Cls (Clear Screen) は、画面を消去する命令、
　２行目の Print は、文字や数値を表示する命令です。

Step02　変数を使用して、値を表示するプログラム

```
Cls
A=35
Print A
```

※変数とは、数値や文字を記憶することができる入れ物です。
※変数名はアルファベットを先頭とした数字との組み合わせなら何でも構いません。
　２行目の A=35 で、数値 35 を変数 A に代入しています。
　３行目の Print A で、変数 A の値を表示しています。
※Print ″A″ とすると、A という文字そのものが表示され、
　Print A とすると、変数 A に代入されている値が表示されます。

Step03　計算をして、結果を表示するプログラム

[時間調整]時間がない場合は行わなくてもよい。

```
Cls
A=15
B=10
Print A+B
```

※プログラムで四則演算を行う場合、＋ － × ÷ が、それぞれ ＋ － ＊ ／ となります。

Step04　プログラム実行後に値を入力させるプログラム

```
Cls
Input ”値を入力してください”,A
Print A
```

※２行目の Input が、入力させる命令です。
　Input "表示させる文字" , 変数

プログラムの解説と実行確認（約 30 分）

練習問題

問題１　底辺と高さを Input 命令で入力して、三角形の面積を表示してみよう。

```
Cls
Input ″底辺を入力してください″, A　　…　底辺
Input ″高さを入力してください″, B　　…　高さ
Print A*B/2　　…　面積の計算
```

※できるだけ、いきなり答えを言わないで、ヒントで解答に導くとよいです。
※Step03 を行っていない場合は、計算式の説明をしてください。

練習問題（約 15 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
1	BASIC プログラムの入力と実行 ①文字の表示 ②変数の使用 ③計算 ④値の入力	①プログラムの入力と実行ができる。 ②変数と計算式が理解できる。 ③Input 命令を理解し使用できる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

評価シート（例）

番号	氏名	知識・技術 ①プログラムの入力と実行ができる。			知識・技術 ②変数と計算式が理解できる。			知識・技術 ③Input 命令を理解し使用できる。			思考・判断・表現 ①目的のプログラムを自ら考え表現(作成)することができる。			主体的に学習に取り組む態度 ①プログラミングに主体的に取り組もうとしている。		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
1	愛知　太郎		A			A			A			A			A	
2	石川　花子		A			B			B			C			B	
3	岩手　次郎		A			A			B			B			A	

Ａ：よくできる　Ｂ：できる　Ｃ：あまりできない

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　２時間目

プログラムの解説と実行確認（約 20 分）

Step01　前回の確認

```
Cls
Input ”値を入力してください”,A
Print A
```

Step02　条件によって判断させるプログラム

```
Cls
Input ”点数を入力してください”,A
If A<30 Then
  Print ”赤点”
Else
  Print ”合格”
End if
```

※点数が 30 未満のとき ″赤点″ と表示し、それ以外のとき ″合格″ と表示する。

```
If 条件 Then
    条件が満たされているとき実行するプログラム
Else
    条件以外のとき実行されるプログラム
End If
```

※If の終わりは、必ず End If で締めます。
※Else の処理がない場合は省略できます。

練習問題

練習問題

問題１　数値を入力して、50 以上なら″A 班″、それ以外なら″B 班″と表示してみよう。

```
Cls
Input ″数値を入力してください″ , A
If A>=50 Then
  Print ″Ａ班″
Else
  Print ″Ｂ班″
End If
```


練習問題（約 25 分）

問題２　数値を入力して、100 以下のときだけ″範囲内″と表示してみよう。

```
Cls
Input ″数値を入力してください″ , A
If A<=100 Then
  Print ″範囲内″
End If
```

問題３　２つの値、ＡとＢを Input 命令で入力し、大きいほうの値を表示してみよう。

```
Cls
Input ″値Ａを入力してください″ , A
Input ″値Ｂを入力してください″ , B
If A>B Then
  Print A
Else
  Print B
End If
```

※生徒間で進行に差が出やすいため、全員ができる到達目標を定めて、
それ以上はよくできる生徒の応用問題とするとよいです。

応用１　カレンダーの月を考えたとき、月の数を入力して、１より小さければ
″小さいです″、12 より大きければ″大きいです″と表示してみよう。

```
Cls
Input ″月を入力してください″ , Tuki
If Tuki<1 Then
  Print ″小さいです″
End If
If Tuki>12 Then
  Print ″大きいです″
End If
```

応用２　三つの数値を入力して、一番大きい数値を表示してみよう。

```
Cls
Input ″数値１を入力してください″ , A
Input ″数値２を入力してください″ , B
Input ″数値３を入力してください″ , C
If A>B Then
  If A>C Then
     Print A
  Else
     Print C
  End If
Else
  If B>C Then
    Print B
  Else
    Print C
  End If
End If
```

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
2	BASIC プログラムの入力と実行 ①判断命令 If	①プログラムの入力と実行ができる。 ②If 命令を理解し使用できる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　３時間目

プログラムの解説と実行確認（約 20 分）

Step01　前回の確認

```
Cls
Input ”年齢を入力してください”,Nen
If Nen>=18 Then
  Print ”成人”
Else
  Print ”子供”
End if
```

Step02　繰り返しプログラム

```
Cls
For I=1 To 10 Step 1
  Print ”ABC”
Next
```

※最初に I の値を 1 とし（初期値）、10 になるまで繰り返す（終了値）
　1回繰り返すたびに 1 ずつ I の値を増やす（増分値）
※ ″ABC″ が 10 回表示されます。

Step03　値を表示する

```
Cls
For I=1 To 10 Step 1
  Print I
Next
```

※ I の値を表示している。I の値の変化が見て取れる。

Step04　While 命令

```
Cls
I=1
While I<=10
  Print I
  I=I+1
Wend
```

※Step03 と同じプログラムを While で記述

練習問題（約 25 分）

練習問題

問題１　For ～ Next を使って、″ABC″ を 50 回表示しなさい。

```
Cls
For A=1 To 50 Step 1
  Print ″ABC″
Next
```

問題２　１～30 までの数値を表示しなさい。

```
Cls
For A=1 To 30 Step 1
  Print A
Next
```

```
Cls
A=1
While A<=30
  Print A
  A=A+1
Wend
```

問題３　１～100 までの奇数を表示しなさい。

```
Cls
For A=1 To 100 Step 2
  Print A
Next
```

```
Cls
A=1
While A<=100
  Print A
  A=A+2
Wend
```

問題４　１～100 までの偶数を表示しなさい。

```
Cls
For A=2 To 100 Step 2
  Print A
Next
```

```
Cls
A=2
While A<=100
  Print A
  A=A+2
Wend
```

応用１　数値を入力して、その数値の段の九九を表示しなさい。

応用２　１～100 までの合計を計算して表示しなさい。

```
Cls
Input ″段は？″;Dan
For A=1 To 9 Step 1
  Print Dan*A
Next
```

```
Cls
Kei=0
For A=1 To 100 Step 1
  Kei=Kei+A
Next
Print Kei
```

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
3	BASIC プログラムの入力と実行 ①繰り返し命令 For ②繰り返し命令 While	①For 命令を理解し使用できる。 ②While 命令を理解し使用できる。	①For と While の違いを理解し使い分けられる。 ②目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　４時間目

プログラムの解説と実行確認（約 20 分）

Step01　前回の確認

```
Cls
For I=1 To 10 Step 1
  Print I
Next
```

```
Cls
I=1
While I<=10
  Print I
  I=I+1
Wend
```

Step02　ランダム値

```
Cls
Print Rnd
```

※０以上１未満の適当な数値（ランダム値）が表示されます。
※ゲーム等の作成には必要な値で、Rnd を計算によって目的の値に変化させます。

（例）	計算式	得られる値
元の式	R=Rnd	0 ～ 0.9999999
10 倍する	R=Rnd*10	0 ～ 9.999999
少数以下を切り捨てる	R=Int(Rnd*10)	0 ～ 9
＋１する	R=Int(Rnd*10)+1	1 ～ 10

※　式　R = Int(Rnd * 10) + 1 により　１～１０までのランダムが得られる。
※　式中の *10 を変えることで、　１～目的の値までのランダムに変えられる。
※　このままでは、毎回ランダムな数値が同じ順番でででるため、プログラムの最初にランダムの初期化 Randomize を入れると良い。

Step03　サイコロを 10 回出力

```
Randomize
Cls
For I=1 To 10 Step 1
  Print Int(Rnd*6)+1
Next
```


練習問題（約 25 分）

練習問題

問題１　１～３のランダムな整数が表示されるプログラムを作成しなさい。

```
Randomize
Cls
Print Int(Rnd*3)+1
```

問題２　１～３までのランダムな整数を発生させ、値が１であれば”グー”、２であれば”パー”、３であれば”チョキ”と表示しなさい。

```
Randomize
Cls
R=Int(Rnd*3)+1
If R=1 Then
  Print "グー"
End If
If R=2 Then
  Print "パー"
End If
If R=3 Then
  Print "チョキ"
End If
```

応用１　サイコロを 1000 回振って、１が出た回数を数えてみよう。

```
Randomize
Cls
Kai=0
For A=1 To 1000 Step 1
   R=Int(Rnd*6)+1
   If R=1 Then
      Kai=Kai+1
   End If
Next
Print "１の回数は ";Kai;" 回です。"
```

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
4	BASIC プログラムの入力と実行 ①ランダム Rnd	①Rnd 命令を理解し使用できる。	①計算によって目的の値を導き出せる。 ②目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕　５時間目

プログラムの解説と実行確認

Step01　前回の確認

```
Randomize
Cls
For I=1 To 10 Step 1
  Print Int(Rnd*6)+1
Next
```

Step02　グラフィック画面の設定

```
GScreen(600,400)
Cls 2
```

※新しいグラフィックを表示するためのウィンドゥが表示される。
※ウィンドゥサイズは 横 600 ドット×縦 400 ドット
※Cls 2 でグラフィック画面を消去

Step03　直線と四角形を描く

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0
```

※座標(0,0)から(300,200)まで、0 番の色(黒)で線を引いている。

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0,B
```

※最後に ,B を付けると四角形となる。(Box)

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0,BF
```

※最後に ,BF を付けると塗りつぶし四角形となる。(Box Fill)

色番号

色番号	名前	色番号	名前	色番号	名前	色番号	名前
0	Black	1	Navy	2	Green	3	Teal
4	Maroon	5	Purple	6	Olive	7	Silver
8	Gray	9	Blue	10	Lime	11	Cyan
12	Red	13	Fuchsia	14	Yellow	15	White

Step04　円を描く

```
GScreen(600,400)
Cls 2
Circle(300,200),100,0
```

※座標(300,200)を中心点とする半径 100 の円を 色番号０(黒)で描く。

```
GScreen(600,400)
Cls 2
Circle(300,200),100,0
```

※座標(300,200)を中心点とする半径 100 の円を 色番号０(黒)で描く。
※Circle 命令では開始角度、終了角度を指定して円弧、比率を変えることで楕円を描くこともできる。

Circle(横座標，縦座標)，半径，色番号，開始角度，終了角度，比率，[F]

```
GScreen(600,400)
Cls 2
Circle(300,200),100,0,,,,F
```

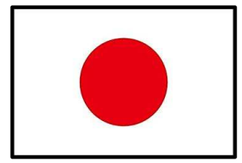
※色番号０(黒)の後に ， を４つ付けて F とすると塗りつぶしもできる。

プログラムの解説と実行確認（約 20 分）

練習問題

問題１　日本の国旗を描いてみよう。

```
GScreen(600, 400)
Cls 2
Line(0, 0)-(600, 400), 15, BF
Circle(300, 200), 100, 12, , , F
```



問題２　フランス国旗を描いてみよう。

```
GScreen(600, 400)
Cls 2
Line(0, 0)-(200, 400), 9, BF
Line(201, 0)-(400, 400), 15, BF
Line(401, 0)-(600, 400), 12, BF
```

応用１　スイス国旗を描いてみよう。

```
GScreen(600, 400)
Cls 2
Line(0, 0)-(600, 400), 12, BF
Line(250, 50)-(350, 350), 15, BF
Line(150, 150)-(450, 250), 15, BF
```

練習問題（約 25 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
5	BASIC プログラムの入力と実行 ①グラフィック命令 直線・四角形・円の描画	①各命令を理解し使用できる。 ②座標を理解している。	①目的の図形を描くプログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

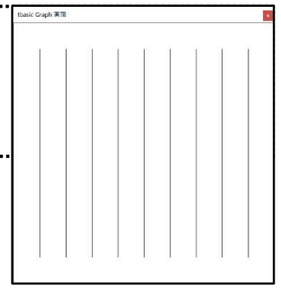
水産海洋技術〔グラフィックプログラミング〕　６時間目

Step01　前回の確認

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),12
Line(300,200)-(350,250),9,B
Line(350,250)-(400,300),2,BF
Circle(300,200),150,14,,,,F
```

Step02　繰り返し命令 For を使って直線を描く

```
GScreen(500,500)
Cls 2
For X=50 To 450 Step 50
  Line(X,50)-(X,450),0
Next
```



※横軸を 50 から 450 まで 50 間隔でずらして描いている。

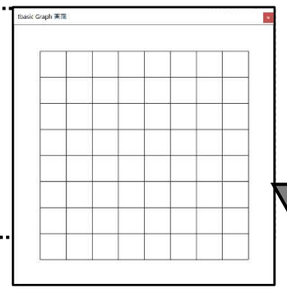
※Step の間隔を変えることで、線の幅が変わります。

プログラムの解説と実行確認（約 20 分）

練習問題

問題１　続けて横線を描いてマス目にしてみよう。

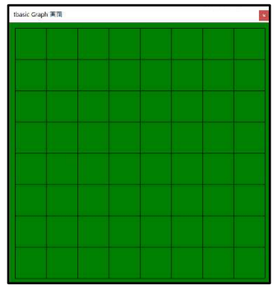
```
GScreen(500,500)
Cls 2
For X=50 To 450 Step 50
  Line(X,50)-(X,450),0
Next
For Y=50 To 450 Step 50
  Line(50,Y)-(450,Y),0
Next
```



練習問題

問題１　背景を色番号２（緑）にし、黒色の線（色番号０）で
８×８のオセロ盤を描いてみよう。ただし、四隅の余白は１０ドットとする。

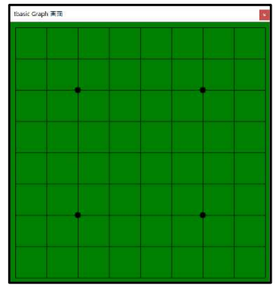
```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For X=10 To 490 Step 60
    Line(X,10)-(X,490),0
Next
For Y=10 To 490 Step 60
    Line(10,Y)-(490,Y),0
Next
```



練習問題

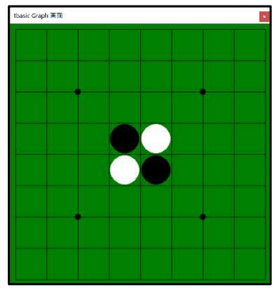
問題２　オセロ盤の中央４×４マスの四隅に、黒い丸を描いてみよう。

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For X=10 To 490 Step 60
    Line(X,10)-(X,490),0
Next
For Y=10 To 490 Step 60
    Line(10,Y)-(490,Y),0
Next
Circle(130,130),6,0,,,,F
Circle(370,130),6,0,,,,F
Circle(130,370),6,0,,,,F
Circle(370,370),6,0,,,,F
```



問題３　オセロ盤に最初の駒を描いてみよう。

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For X=10 To 490 Step 60
    Line(X,10)-(X,490),0
Next
For Y=10 To 490 Step 60
    Line(10,Y)-(490,Y),0
Next
Circle(130,130),6,0,,,,F
Circle(370,130),6,0,,,,F
Circle(130,370),6,0,,,,F
Circle(370,370),6,0,,,,F
Circle(220,220),28,0,,,,F
Circle(280,220),28,15,,,,F
Circle(220,280),28,15,,,,F
Circle(280,280),28,0,,,,F
```



応用１　オセロゲームを進めてみよう。

練習問題（約 25 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
6	BASIC プログラムの入力と実行 ①グラフィック命令 直線・四角形・円を使って オセロ盤の描画	①各命令を理解し使用できる。 ②座標を理解している。	①オセロ盤を描くために、各命令を組み合わせてプログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

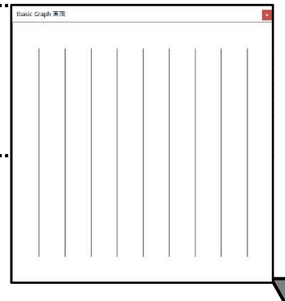
愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕

７時間目

Step01　前回の確認

```
GScreen(500,500)
Cls 2
For X=50 To 450 Step 50
  Line(X,50)-(X,450),0
Next
```



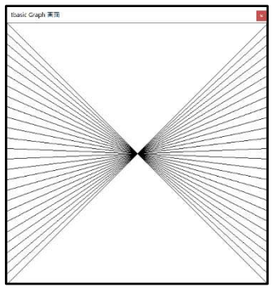
プログラムの解説と実行確認（約10分）

練習問題

問題１　次のような線を、繰り返し命令 For を使って描きなさい。
ただし、線の間隔は 20 ドットとする。

```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(500,Y),0
Next
```

問題２　次のような線を、繰り返し命令 For を使って描きなさい。

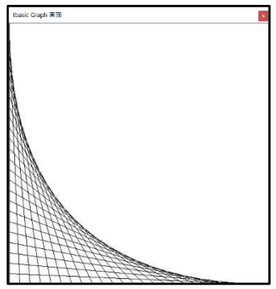


```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(500,500-Y),0
Next
```

※始点と終点の座標がどのように動いていくかを説明して生徒に考えさせるとよいです。

練習問題

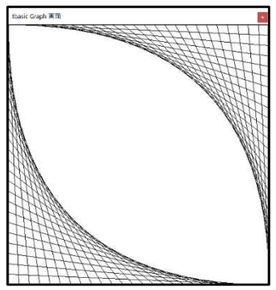
問題３　次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(Y,500),0
Next
```

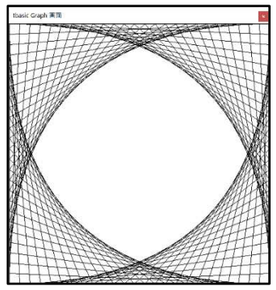
※始点と終点の座標がどのように動いていくかを説明して生徒に考えさせるとよいです。

応用１　次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(Y,500),0
Next
For Y=0 To 500 Step 20
  Line(Y,0)-(500,Y),0
Next
```

応用２　次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(Y,500),0
Next
For Y=0 To 500 Step 20
  Line(Y,0)-(500,Y),0
Next
For Y=0 To 500 Step 20
  Line(0,Y)-(500-Y,0),0
Next
For Y=0 To 500 Step 20
  Line(Y,500)-(500,500-Y),0
Next
```

練習問題（約35分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
7	BASIC プログラムの入力と実行 ①グラフィック命令 繰り返し命令を使って、様々な直線を描く	①繰り返し命令を理解し使用できる。 ②座標を理解している。	①図形を描くために、プログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

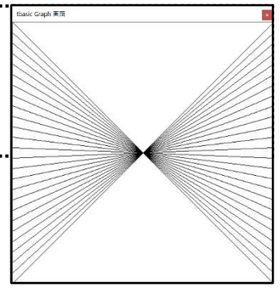
水産海洋技術〔プログラミング〕

８時間目

プログラムの解説と実行確認

Step01　前回の確認

```
GScreen(500,500)
Cls 2
For Y=0 To 500 Step 20
  Line(0,Y)-(500,500-Y),0
Next
```



Step02　Read と Data

```
Cls
Read A
Print A
Data 100
```

※Read 命令は、Data の先頭から順番に値を読み込む命令である。

```
Cls
Read A,B,C
Print A;B;C
Data 100,200,300
```

```
Cls
For I=1 To 8 Step 1
  Read A
  Print A
Next
Data 50,30,75,90,27,10,100,60
```

※繰り返し命令 For を使って、連続して読み込むこともできる。

```
Cls
For I=1 To 8 Step 1
  Read A
  Print ”点数は”;A;”点”
Next
Data 50,30,75,90,27,10,100,60
```

※Print 命令に ； を付けると、表示を連続させることができる。

プログラムの解説と実行確認（約 30 分）

```
Cls
MX=0
For I=1 To 8 Step 1
  Read A
  Print ”点数は”;A;”点”
  If MX<=A Then
    MX=A
  End If
Next
Print ”最高点は”;MX;”点”
Data 50,30,75,90,27,10,100,60
```

※最初に、変数 MX に最も小さい値を入れておく。
Read 命令で値を読むごとに、MX と比較して、読み込んば値の方が大きければ、
その値を MX に代入する。これを繰り返すことによって、最大値を MX に記憶させる
ことができる。

練習問題（約 15 分）

練習問題

問題１　15 人分のテストの点数を Data に入力し、Read 命令で表示しなさい。

```
Cls
MX=0
For I=1 To 15 Step 1
   Read A
   Print ”点数は”;A;”点”
Next
Data 50,30,75,90,27,10,100,60,49,84,61,18,50,97,77
```

問題２　最低点を表示しなさい。

```
Cls
MN=999
For I=1 To 15 Step 1
   Read A
   Print ”点数は”;A;”点”
   If MN>=A Then
      MN=A
   End If
Next
Print ”最低点は”;MN;”点”
Data 50,30,75,90,27,10,100,60,49,84,61,18,50,97,77
```

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
8	BASIC プログラムの入力と実行 ①Read・Data 命令 Data から値を読み込み処理を行う	①Read 命令を理解し使用することができる。 ②繰り返し命令 For、判断命令 If と組み合わせて使用することができる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕

９時間目

プログラムの解説と実行確認（約30分）

Step01　前回の確認

```
Cls
For I=1 To 5 Step 1
  Read A
  Print A
Next
Data 50,30,75,90,27
```

Step02　一次元配列

```
Cls
Dim A(5)
A(1)=50
A(2)=30
A(3)=75
A(4)=90
A(5)=27
Print A(3)
```

※Dim を使用して配列を使用することを宣言する。
変数Ａの１番目A(1)に 50 を代入、A(2)に 30 を代入、A(3)に 75 を代入、A(4)に 90 を代入、A(5)に 27 を代入して、最後にA(3)を表示している。
このように、変数名に添字を付けて使用する変数を配列という。

```
Cls
Dim A(5)
For I=1 To 5 Step 1
  Read A(I)
Next
Data 50,30,75,90,27
Print A(3)
```

※繰り返し命令 For と組み合わせることによって、配列に一気に値を入れることができる。

```
Cls
Dim A$(5)
For I=1 To 5
  Read A$(I)
Next
Data ”A”,”B”,”C”,”D”,”E”

For I=5 To 1 Step -1
  Print A$(I)
Next
```

※文字を配列 A$に記憶し、それを逆順で表示している。文字を記憶するときの変数名には$を付ける。

練習問題（約15分）

練習問題

問題１　８個の数値を配列Ａに読み込みなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
   Read A(I)
Next
Data 50,30,75,90,27,100,40,68
```

問題２　続けて、読み込んだ数値をすべて表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
   Read A(I)
Next
Data 50,30,75,90,27,100,40,68

For I=1 To 8 Step 1
   Print A(I)
Next
```

応用１　読み込んだ数値の基数番目を表示しなさい。また偶数番目も表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
   Read A(I)
Next
Data 50,30,75,90,27,100,40,68

For I=1 To 8 Step 2
   Print A(I)
Next
```

```
Cls
Dim A(8)
For I=1 To 8 Step 1
   Read A(I)
Next
Data 50,30,75,90,27,100,40,68

For I=2 To 8 Step 2
   Print A(I)
Next
```

応用２　読み込んだ数値を逆順にすべて表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
   Read A(I)
Next
Data 50,30,75,90,27,100,40,68

For I=8 To 1 Step -1
   Print A(I)
Next
```

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
9	BASIC プログラムの入力と実行 ①一次元配列 配列を使った処理を行う	①Read 命令を理解し配列に値を読み込むことができる。 ②繰り返し命令を組み合わせて使用することができる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕

10 時間目

プログラムの解説と実行確認

Step01　Read ・ Data の確認

```
Cls
For I=1 To 5 Step 1
  Read A
  Print A
Next
Data 50,30,75,90,27
```

Step02　二重ループ

```
Cls
For Y=1 To 4 Step 1
  For X=1 To 4 Step 1
    Read A
    Print A;
  Next
  Print
Next
Data  1, 2, 3, 4
Data  5, 6, 7, 8
Data  9,10,11,12
Data 13,14,15,16
```

※二重ループの動作を確認します。
※内ループの Print は ； を付けて、横につなげていますが、内ループを抜けた段階で、
　； を付けずに改行しています。

Step03　掛け算九九

```
Cls
For Y=1 To 9 Step 1
  For X=1 To 9 Step 1
    Print X*Y;
  Next
  Print
Next
```


Step04　キャラクターの作成

```
Cls
For Y=1 To 4 Step 1
  For X=1 To 4 Step 1
    Read C
    If C=1 Then
      Print "■";
    Else
      Print "　";
    End If
  Next
  Print
Next
Data 1,1,1,1
Data 1,1,0,1
Data 1,0,1,1
Data 1,1,1,1
```

※Data の値によって、表示する文字を変えている。
※Print の"■"と"　"は全角文字と空白です。

プログラムの解説と実行確認（約 30 分）

練習問題

問題１　８×８のキャラクターにしてみよう。

```
Cls
For Y=1 To 8 Step 1
   For X=1 To 8 Step 1
      Read C
      If C=1 Then
         Print "■";
      Else
         Print "　";
      End If
   Next
   Print
Next
Data 0,1,0,0,0,0,1,0
Data 0,1,0,0,0,0,1,0
Data 0,1,1,0,0,1,1,0
Data 1,1,1,1,1,1,1,1
Data 1,0,0,1,1,0,0,1
Data 1,0,1,1,1,1,0,1
Data 1,1,1,1,1,1,1,1
Data 0,1,1,1,1,1,1,0
```

問題２　キャラクターを自由に変えてみよう。

練習問題（約 15 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
10	BASIC プログラムの入力と実行 ①二重ループ 　二重ループを使った処理を行う	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、目的の結果を得るためのプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕　11 時間目

Step01　キャラクター表示をグラフィックス化する

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    If C=1 Then
      Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),0,BF
    End If
  Next
Next
Data 0,1,0,0,0,0,1,0
Data 0,1,0,0,0,0,1,0
Data 0,1,1,0,0,1,1,0
Data 1,1,1,1,1,1,1,1
Data 1,0,0,1,1,0,0,1
Data 1,0,1,1,1,1,0,1
Data 1,1,1,1,1,1,1,1
Data 0,1,1,1,1,1,1,0
```

色

※Read C で読み込んだ Data の値が 1 ならば、Print "■"ではなく、
Line 命令によって、縦横 20 ドットの塗りつぶし四角形を描いている。

※色の部分を黒色の 0 ではなく、C とすることで、Data 内の値を色番号とすることができる。

Step02　色付きのキャラクターを表示する

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),C,BF
  Next
Next
Data 15,15,12,12,12,12,12,12
Data 15, 9, 6, 9, 6,14,14,15
Data  6,15, 6, 6, 6, 6, 6,15
Data  8,15, 6,14, 6, 6,15, 6
Data  8,12,12,12,12,12,12, 8
Data 15,15,12,12,12,12,15,15
Data 15,15, 8, 8, 8, 8,15,15
Data 15, 8, 8,15,15, 8, 8,15
```

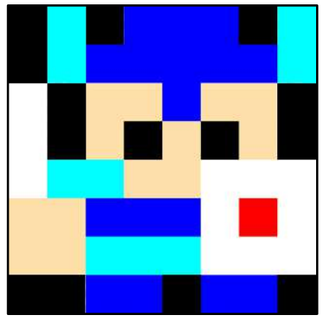
プログラムの解説と実行確認（約 30 分）

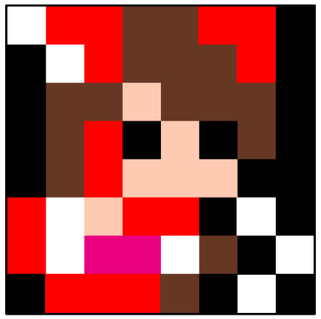
色番号

色番号	名前	色番号	名前	色番号	名前	色番号	名前
0	Black	1	Navy	2	Green	3	Teal
4	Maroon	5	Purple	6	Olive	7	Silver
8	Gray	9	Blue	10	Lime	11	Cyan
12	Red	13	Fuchsia	14	Yellow	15	White

練習問題

問題１　好きなキャラクターを描いてみよう。





練習問題（約 15 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
11	BASIC プログラムの入力と実行 ①二重ループ 　二重ループを使ってキャラクターの描画	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、キャラクターを描画するプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕　12 時間目

Step01　データを工夫して処理しやすくする

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  Read C$ ←――― 1 行分の文字列として読み込む
  For X=1 To 8 Step 1
    C=Val(”&H”+Mid$(C$,X,1))
    Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),C,BF
  Next
Next
Data ”FFCCCCCC”
Data ”F9696EEF”
Data ”6F66666F”
Data ”8F6E66F6”
Data ”8CCCCCC8”
Data ”FFCCCCFF”
Data ”FF8888FF”
Data ”F88FF88F”
```

色データを 16 進数 1 桁にする。
データが、数値から文字になるため″ ″で囲う

C = Val(″&H″ + Mid$(C$, X , 1))
文字列 C$の X 番目から 1 文字抜き出している。
Val(″&H″ は 16 進を 10 進にするため。

※少し難易度が高いため、「このようにすれば、16 進 1 桁を色番号として、キャラクターが描ける。」程度の説明で良い。

プログラムの解説と実行確認（約 20 分）

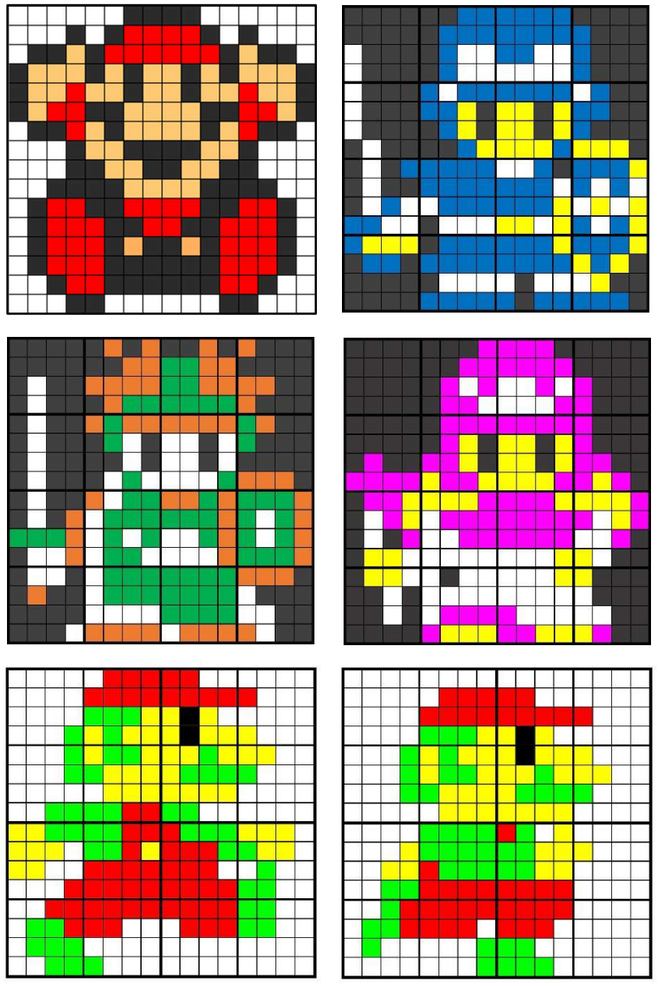
練習問題

問題１　キャラクターサイズを 16×16 にしてみよう。

```
GScreen(500,500)
Cls 2
For Y=1 To 16 Step 1
  Read C$
  For X=1 To 16 Step 1
    C=Val(”&H”+Mid$(C$,X,1))
    Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),C,BF
  Next
Next
Data ”FFFFFFFFFFFFFFFF”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”F000000000000000F”
Data ”FFFFFFFFFFFFFFFF”
```

プログラムの解説と実行確認

問題２　好きなキャラクターを描いてみよう。



練習問題（約 25 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
12	BASIC プログラムの入力と実行 ①二重ループ 二重ループを使ってキャラクターの描画	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、キャラクターを描画するプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕　13 時間目

プログラムの解説と実行確認（約 25 分）

Step01　ランダムな位置に円を描く

```
GScreen(500,500)
Cls 2
Randomize

X=Int(Rnd*500)
Y=Int(Rnd*500)
Circle(X,Y),30,0
```

ランダムな位置に円を描いている。

Step02　サブルーチン

```
GScreen(500,500)
Cls 2
Randomize
Call MARU()
End
```

← Call 命令で、サブルーチンを呼び出す

```
Sub MARU()
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,0
End Sub
```

ランダムな位置に円を描くサブルーチン

※独立したプログラムのかたまりを、サブルーチンという。
※Call 命令で、サブルーチンを呼び出すことができる。

```
GScreen(500,500)
Cls 2
Randomize
For I=1 To 10 Step 1
  Call MARU()
Next
End
```

```
Sub MARU()
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,0
End Sub
```

※サブルーチンは何度でも呼び出して実行することができる。

練習問題（約 25 分）

練習問題

問題１　ランダムな位置に、ランダムな大きさの四角形を描くプログラムを作成しなさい。

```
GScreen(500,500)
Cls 2
Randomize

X1=Int(Rnd*500)
Y1=Int(Rnd*500)
X2=Int(Rnd*500)
Y2=Int(Rnd*500)
Line(X1,Y1)-(X2,Y2),0,B
```

問題２　ランダムな大きさの四角形を描くプログラムをサブルーチン化しなさい。

```
GScreen(500,500)
Cls 2
Randomize
Call SIKAKU()
End

Sub SIKAKU()
  X1=Int(Rnd*500)
  Y1=Int(Rnd*500)
  X2=Int(Rnd*500)
  Y2=Int(Rnd*500)
  Line(X1,Y1)-(X2,Y2),0,B
End Sub
```

問題３　ランダムな大きさの四角形を描くプログラムをサブルーチンを１０回実行しなさい。

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
13	BASIC プログラムの入力と実行 ①サブルーチン 何度も行う処理をサブルーチン化する	①サブルーチンを理解し、プログラミングすることができる。	①サブルーチンの動作を理解し、プログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔グラフィックプログラミング〕　14 時間目

プログラムの解説と実行確認（約 35 分）

Step01　前回の確認

```
GScreen(500,500)
Cls 2
Randomize
Call MARU()
End
```

← Call 命令で、サブルーチンを呼び出す

```
Sub MARU()
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,0
End Sub
```

ランダムな位置に円を描くサブルーチン

Step02　円の色を指定するために、値を渡す

```
GScreen(500,500)
Cls 2
Randomize
Call MARU(12)
End
```

← カッコ内に渡す値を記述する

```
Sub MARU(C)
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,C
End Sub
```

カッコ内に書いた数値を、変数 C で受け取る。

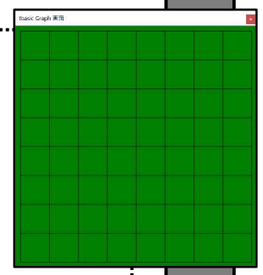
Step03　オセロ盤を描いて、コマを表示するサブルーチンを作成する

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For A=10 To 490 Step 60
  Line(A,10)-(A,490),0
  Line(10,A)-(490,A),0
Next
Call KOMA(4,4, 0)
Call KOMA(5,5, 0)
Call KOMA(5,4,15)
Call KOMA(4,5,15)
End
```

オセロ盤を描いている

黒コマを置いている

白コマを置いている



```
Sub KOMA(X,Y,C)
  Circle(X*60-20,Y*60-20),28,C,,,,F
End Sub
```

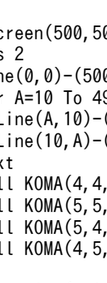
コマを表示するサブルーチン

X:横　Y:縦　C:コマの色

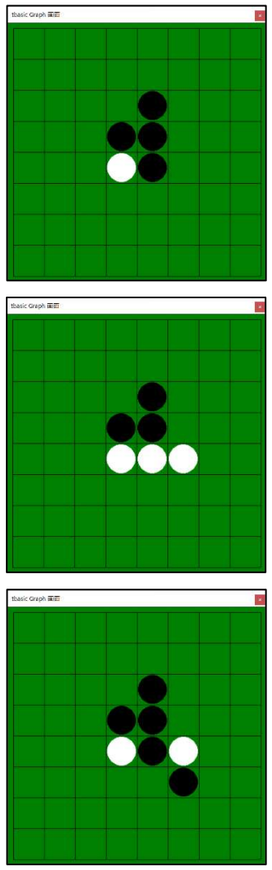
練習問題

問題１　オセロを進行させてみよう。

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For A=10 To 490 Step 60
  Line(A,10)-(A,490),0
  Line(10,A)-(490,A),0
Next
Call KOMA(4,4, 0)
Call KOMA(5,5, 0)
Call KOMA(5,4,15)
Call KOMA(4,5,15)

Call KOMA(5,3, 0)
Call KOMA(5,4, 0)

Call KOMA(6,5,15)
Call KOMA(5,5,15)

Call KOMA(6,6, 0)
Call KOMA(5,5, 0)
```



```
End

Sub KOMA(X,Y,C)
  Circle(X*60-20,Y*60-20),28,C,,,,F
End Sub
```



練習問題（約 15 分）

評価のポイント

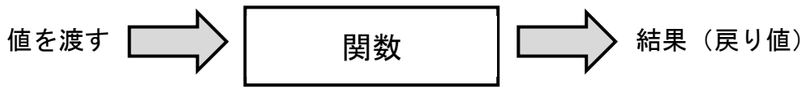
時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
14	BASIC プログラムの入力と実行 ①サブルーチン 値の受け渡しをするサブルーチン	①値を受け渡すサブルーチンを理解しプログラミングすることができる。	①サブルーチンの動作を理解し、オセロプログラムに例えて値の受け渡しを行うプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　15 時間目

■ 関数の作成

関数とは、値を渡すと、結果（戻り値）が得られるサブルーチンです。



Step01　関数の作成

テストの点数を渡すと、不合格(30 点未満)ならば０、合格(それ以外)ならば１の値を返す関数を作成する。

```
Cls
A=TESUTO(29)                    ※テストの点数は 29 点
Print A
End

Function TESUTO(TEN)            ※テストの点数を変数 TEN で受け取る。
  If TEN<30 Then
    K=0
  Else
    K=1
  End If
  TESUTO=K                      ※返す値を設定する
End Function
```

帰ってきた値をもとに、合格・不合格を文字で表示する。

```
Cls
A=TESUTO(29)
If A=0 Then
  Print ”不合格”
Else
  Print ”合格”
End If
End

Function TESUTO(TEN)
  If TEN<30 Then
    K=0
  Else
    K=1
  End If
  TESUTO=K
End Function
```

プログラムの解説と実行確認（約 25 分）

練習問題

問題１　テストの点数を渡すと、優秀(80 点以上)なら１、それ以外なら２の値を返す関数を作成しなさい。

```
Cls
A=TESUTO(80)
Print A
End

Function TESUTO(TEN)
  If TEN>=80 Then
    K=1
  Else
    K=2
  End If
  TESUTO=K
End Function
```

問題２　テストの点数を渡すと、優秀(80 点以上)なら１、再考査(30 点未満)なら０、それ以外なら２の値を返す関数を作成しなさい。

```
Cls
A=TESUTO(80)
Print A
End

Function TESUTO(TEN)
  If TEN>=80 Then
    K=1
  Else
    If TEN<30 Then
      K=0
    Else
      K=2
    End If
  End If
  TESUTO=K
End Function
```

練習問題（約 25 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
15	BASIC プログラムの入力と実行 ①関数 関数の作成と利用	①値を受け渡し、処理した結果を返す関数を理解しプログラミングすることができる。	①関数の動作を理解し、簡単な関数を使ったプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔プログラミング〕　16 時間目

■ 二次元配列

表のような構造を持つ配列です。
一次元配列同様、Dim 命令で使用を宣言します。

（例１）　Dim A(5,3)　　※横５マス、縦３マスの箱を用意する。

変数Ａ

(1,1)	(2,1)	(3,1)	(4,1)	(5,1)
(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)

値を代入する際には、　A(1,1)=10 ： A(2,1)=20 ： A(3,1)=30 の様にする。

（例２）　３人分の、５教科のテストの点数を記憶する。

変数 TEN

	国語	社会	数学	理科	英語
１人目	65	81	34	42	16
２人目	80	95	77	70	75
３人目	100	100	98	95	92

```
Cls
Dim TEN(5,3)
TEN(1,1)=65 : TEN(2,1)=81 : TEN(3,1)=34 : TEN(4,1)=42 : TEN(5,1)=16
TEN(1,2)=80 : TEN(2,2)=95 : TEN(3,2)=77 : TEN(4,2)=70 : TEN(5,2)=75
TEN(1,3)=100: TEN(2,3)=100: TEN(3,3)=98 : TEN(4,3)=95 : TEN(5,3)=92
```

としてもよいが、
For で繰り返し、Read、Data、で値を読み込んで代入させると便利である。

```
Cls
Dim TEN(5,3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J,I)
  Next
Next
Data  65, 81, 34, 42, 16
Data  80, 95, 77, 70, 75
Data 100,100, 98, 95, 92
```

プログラムの解説と実行確認

このままでは、何も表示されず、結果がわかりにくいため、中身を表示する
プログラムを追加する。

```
Cls
Dim TEN(5,3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J,I)
  Next
Next

For I=1 To 3
  For J=1 To 5
    Print TEN(J,I);
  Next
  Print
Next

Data  65, 81, 34, 42, 16
Data  80, 95, 77, 70, 75
Data 100,100, 98, 95, 92
```

表示するプログラム

プログラムの解説と実行確認（約 30 分）

練習問題

問題１　横４マス、縦２マスの二次元配列を作り、以下の様な値を代入して、
表示させてみよう。

50	65	25	80
100	40	75	95

応用１　この中から一番大きい値を表示してみよう。

```
Cls
Dim TEN(5,3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J,I)
  Next
Next
MX=0
For I=1 To 3
  For J=1 To 5
    If TEN(J,I)>MX Then
      MX=TEN(J,I)
    End If
  Next
Next
Print ”最大値：”;MX
Data  65, 81, 34, 42, 16
Data  80, 95, 77, 70, 75
Data 100,100, 98, 95, 92
```

練習問題（約 25 分）

評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
16	BASIC プログラムの入力と実行 ①二次元配列 二次元配列の作成と利用	①二次元配列について理解しプログラミングすることができる。	①二次元配列を理解し、簡単なプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔ゲームプログラミング〕　No.01

■ キー入力　Inkey$

どのボタンが押されているかを判断するために、キー入力操作の仕方を学びます。
プログラム実行後に、文字や数値を入力する命令に Input がありましたが、
Input は、入力されるまでプログラムが停止するため、ゲーム等でキャラクターを動かしたり
するには、使いにくいです。ここで使用する命令が、Inkey$ です。

（使い方）　A$=Inkey$　　命令実行時に押されているキーが A$に代入される。

Step01　キー入力　Inkey$

```
Cls
A$=Inkey$
```

実行しても、プログラムが一瞬で終了してしまいます。
Inkey$ は、プログラムがその場で停止しないため、工夫が必要です。

Step02　キー入力の工夫

```
Cls
A$=""
While(A$="")
  A$=Inkey$
WEnd
```

While 命令を使用し、A$ が空っぽ(何も入力されていない)間、繰り返すようにします。
ボタンが押されると、A$ に文字が代入されるため、繰り返しを終了します。

Step03　入力された文字を表示する

```
Cls
A$=""
While(A$="")
  A$=Inkey$
WEnd
Print A$
```

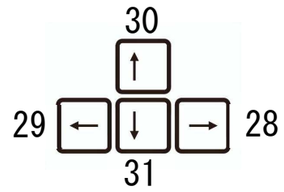
Print A$ とすれば、代入された文字を表示できます。
しかし、矢印キーや、Enter キー、ESC キーなど、特殊なキーの表示ができません。
特殊なキーが押されたとき、それを判断する場合には、キーコードに変換して処理します。

Step04　キーコードで表示する

```
Cls
A$=""
While(A$="")
  A$=Inkey$
WEnd
Print ASC(A$)
```

ASC( ) で囲うと、文字を文字コードに変換して表示できます。

これを応用して、カーソルキーのキーコードを調べて判断することができます。



※生徒に調べさせて覚えておくよう指示するとよいです。

Step05　キーコードで表示する

```
Cls
A$=""
While(A$="")
  A$=Inkey$
WEnd
If ASC(A$)=28 Then Print "右"
```

右カーソルキーを押すと、"右"と表示されます。

問題　左・上・下 も同様にプログラムしてみよう。

Step06　グラフィック表示してみよう

```
GScreen(400,400)
X=200:Y=200
PSet(X,Y),0
```

画面の中央に黒い点を描きます。　※ PSet(,) は、1 ドット点を描く命令

ST という変数を用意して、0の間繰り返すようにします。

```
GScreen(400,400)
X=200:Y=200
ST=0
While(ST=0)
  PSet(X,Y),0
WEnd
```

繰返しの中でキー入力を行い、Enter キー(13 番)が押されたら、ST を 1 にして繰返しを
終了するようにします。

```
GScreen(400,400)
X=200:Y=200
ST=0
While(ST=0)
  PSet(X,Y),0
  A$=""
  While(A$="")
    A$=Inkey$
  WEnd
  If ASC(A$)=13 Then
    ST=1
  End If
Wend
```

右カーソルキー(28 番)が押されたら、X の値を +1 するようにします。

```
GScreen(400,400)
X=200:Y=200
ST=0
While(ST=0)
  PSet(X,Y),0
  A$=""
  While(A$="")
    A$=Inkey$
  WEnd
  If ASC(A$)=13 Then
    ST=1
  End If
  If ASC(A$)=28 Then
    X=X+1
  End If
WEnd
```

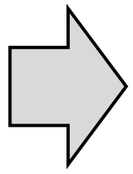
右カーソルキーを押すと、黒の点が右に移動していきます。
また、Enter キーを押すと、プログラムが終了します。

問題　左・上・下 も移動するようにプログラムしてみよう。

ゲーム化します

Step07　自動的に動いていき、カーソルキーで向きを変えられるように変更する

```
GScreen(400,400)
X=200:Y=200
ST=0
While(ST=0)
  PSet(X,Y),0
  A$=""
  While(A$="")
    A$=Inkey$
  WEnd
  If ASC(A$)=13 Then
    ST=1
  End If
  If ASC(A$)=28 Then
    X=X+1
  End If
  If ASC(A$)=29 Then
    X=X-1
  End If
  If ASC(A$)=30 Then
    Y=Y-1
  End If
  If ASC(A$)=31 Then
    Y=Y+1
  End If
WEnd
```



```
GScreen(400,400)
X=200:Y=200:MX=1:MY=0
ST=0
While(ST=0)
  A$=""
  While(A$="")
    A$=Inkey$
    X=X+MX
    Y=Y+MY
    PSet(X,Y),0
  WEnd
  If ASC(A$)=13 Then
    ST=1
  End If
  If ASC(A$)=28 Then
    MX=1:MY=0
  End If
  If ASC(A$)=29 Then
    MX=-1:MY=0
  End If
  If ASC(A$)=30 Then
    MX=0:MY=-1
  End If
  If ASC(A$)=31 Then
    MX=0:MY=1
  End If
WEnd
```



```
GScreen(400,400)
X=200:Y=200:MX=1:MY=0
ST=0
While(ST=0)
  A$=""
  While(A$="")
    A$=Inkey$
    X=X+MX
    Y=Y+MY
    PSet(X,Y),0
  WEnd
  If ASC(A$)=13 Then
    ST=1
  End If
  If ASC(A$)=28 Then
    MX=1:MY=0
  End If
  If ASC(A$)=29 Then
    MX=-1:MY=0
  End If
  If ASC(A$)=30 Then
    MX=0:MY=-1
  End If
  If ASC(A$)=31 Then
    MX=0:MY=1
  End If
WEnd
```

移動方向用の変数を用意します。

何もキーを押していない時に、自動的にその方向へ移動させます。

カーソルキーを押したときは、移動方向を変更します。
MX は横の移動方向
MY は縦の移動方向

Step08　ゲームオーバーを考える

ゲームオーバーとなる場合は、以下の２点があります。
①　自分の線にあたった　（点を描く際、すでに黒色の点が書かれている）
②　画面からはみ出した　（変数ＸとＹが、０より小さい、399 より大きいときゲームオーバーとする）

①は、GetRGBPixel(x,y) 命令によって、その場所の色情報を 16 進数で取得できるため、黒色(#000000)なら "GAME OVER" とします。

```
If GetRGBPixel(x,y)="#000000" Then
   Cls:Print "GAME OVER"
   End
End If
```

②は、ＸとＹの値が、範囲外かどうかで判断します。

```
If X<0 Or X>399 Or Y<0 Or Y>399 Then
   Cls:Print "GAME OVER"
   End
End If
```

①は点を描く前、②点を描いた後に挿入します。

問題　変数 SC を使って、SCORE が最後に表示されるようにしてみよう。

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔ゲームプログラミング〕　No.02

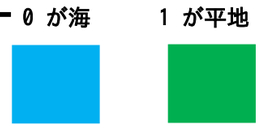
■ マップスクロール

自分でマップを作成し、そのマップを歩くことができるプログラムを作成します。
今回の画面表示は３×３とします。

Step01　マップデータを一次元配列で作成します。

```
GScreen(300,300)
Dim A$(10)
A$( 1)="0000000000"
A$( 2)="0110000110"
A$( 3)="0011111100"
A$( 4)="0001111000"
A$( 5)="0011001110"
A$( 6)="0111001110"
A$( 7)="0011111000"
A$( 8)="0001111110"
A$( 9)="0111000010"
A$(10)="0000000000"
```

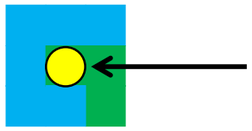
0 が海　　1 が平地



Step02　現在地を設定します。

```
X=2:Y=2
```

Step03　自分を中心に３×３のマップを表示します。



○が自分の位置（X,Y）
その手前から、一つ先までを表示するため
横は X-1～X+1 まで
縦は Y-1～Y+1 までを表示する。

```
For I=-1 To 1
  For J=-1 To 1
    C=Val(Mid$(A$(Y+I)),X+J,1))
    If C=0 Then
      Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),9,BF
    End If
    If C=1 Then
      Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),10,BF
    End If
  Next
Next
```

自分が中心にいるように見せるために、黄色の円を描く。

```
Circle(150,150),49,14,,,,F
```

Step04　キー入力をします。

```
K$=""
While(K$="")
  K$=Inkey$
WEnd
Select Case ASC(K$)
  Case 28:X=X+1
  Case 29:X=X-1
  Case 30:Y=Y-1
  Case 31:Y=Y+1
End Select
```

前回 If 命令で行っていた判断を
Select 命令で記述しました。

以降、マップ表示の部分までを ESC(コード 27)が押されるまで繰り返すようにする。

```
GScreen(300,300)
Dim A$(10)
A$( 1)="0000000000"
A$( 2)="0110000110"
A$( 3)="0011111100"
A$( 4)="0001111000"
A$( 5)="0011001110"
A$( 6)="0111001110"
A$( 7)="0011111000"
A$( 8)="0001111110"
A$( 9)="0111000010"
A$(10)="0000000000"
X=2:Y=2
ST=0
While(ST=0)
  For I=-1 To 1
    For J=-1 To 1
      C=Val(Mid$(A$(Y+I),X+J,1))
      If C=0 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),9,BF
      End If
      If C=1 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),10,BF
      End If
    Next
  Next
  Circle(150,150),49,14,,,,F
  K$=""
  While(K$="")
    K$=Inkey$
  WEnd
  Select Case ASC(K$)
    Case 27:ST=1
    Case 28:X=X+1
    Case 29:X=X-1
    Case 30:Y=Y-1
    Case 31:Y=Y+1
  End Select
WEnd
```

Step05　海の上を歩けないようにする。

キーを押して、ＸまたはＹの値を変化させる前に、
ＢＸ＝Ｘ：ＢＹ＝Ｙ
を実行して、ＸとＹの値をＢＸとＢＹにコピーしておく。

その後、ＸまたはＹの値を変化させた後に、マップを調べて、海ならば
Ｘ＝ＢＸ：Ｙ＝ＢＹ
として、変更前の値に戻すことで、移動していないことにする。

```
BX=X:BY=Y                    ← X と Y の値をコピーしておく
Select Case ASC(K$)
  Case 27:ST=1
  Case 28:X=X+1
  Case 29:X=X-1
  Case 30:Y=Y-1
  Case 31:Y=Y+1
End Select
C=Val(Mid$(A$(Y),X,1))       ← 移動後のマップを調べる
If C=0 Then                  ← 海ならば
  X=BX:Y=BY                     X と Y の値をもとに戻す
End If
```

これで、海に入れなくなります。

問題１　マップを大きくしてみよう。

横に広げる場合は、そのまま数字を増やしてください。

（例）　A$(1)=”00000000000000000000000000000000”

縦に広げる場合は、配列の数を増やして、行を増やしていきます。

```
（例）　Dim A$(30)
        A$(11)=”0111111110”
        A$(12)=”0111111110”
```

問題２　マップを拡張してみよう。

マップデータ中に２を追加し、２のマップは濃い緑（色番号２）で表示する。

マップデータ中に３を追加し、３のマップは灰色（色番号８）で表示する。

（例）

```
A$( 1)=”0000000000”
A$( 2)=”0112000210”
A$( 3)=”0012222100”
A$( 4)=”0001221000”
A$( 5)=”0013001130”
A$( 6)=”0111001110”
A$( 7)=”0011111000”
A$( 8)=”0001113110”
A$( 9)=”0111000010”
A$(10)=”0000000000”
```


完成プログラム

```
GScreen(300,300)
Dim A$(10)
A$( 1)=”0000000000”
A$( 2)=”0112000210”
A$( 3)=”0012222100”
A$( 4)=”0001221000”
A$( 5)=”0013001130”
A$( 6)=”0111001110”
A$( 7)=”0011111000”
A$( 8)=”0001113110”
A$( 9)=”0111000010”
A$(10)=”0000000000”
X=2:Y=2
ST=0
While(ST=0)
  For I=-1 To 1
    For J=-1 To 1
      C=Val(Mid$(A$(Y+I),X+J,1))
      If C=0 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),9,BF
      End If
      If C=1 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),10,BF
      End If
      If C=2 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),2,BF
      End If
      If C=3 Then
        Line((J+1)*100,(I+1)*100)-((J+2)*100,(I+2)*100),8,BF
      End If
    Next
  Next
  Circle(150,150),49,14,,,,F
  K$=””
  While(K$=””)
    K$=Inkey$
  WEnd
  BX=X:BY=Y
  Select Case ASC(K$)
    Case 27:ST=1
    Case 28:X=X+1
    Case 29:X=X-1
    Case 30:Y=Y-1
    Case 31:Y=Y+1
  End Select
  C=Val(Mid$(A$(Y),X,1))
  If C=0 Then
    X=BX:Y=BY
  End If
WEnd
```

愛知県立三谷水産高等学校　教諭　山田　学

水産海洋技術〔ゲームプログラミング〕　No.03

■ 棒倒し法アルゴリズムを使用した迷路ゲーム

教科書 P116～P117 の棒倒し法アルゴリズムを使用した、迷路自動作成ゲームです。

Step01　棒倒しアルゴリズムを使って二次元配列に迷路を作成します。

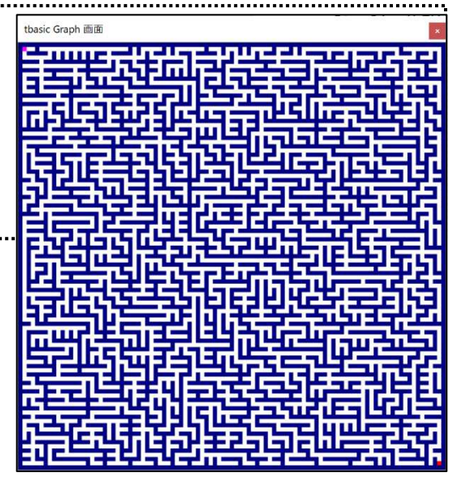
```
Randomize
Gscreen(507,507)
DIM M(101,101)
For I=1 To 101 Step 1
  M(I,1)=1:M(I,101)=1:M(1,I)=1:M(101,I)=1
Next
For Y=3 To 99 Step 2
  For X=3 To 99 Step 2
    M(X,Y)=1
    XT=0:YT=0
    While(M(X+XT,Y+YT)=1)
      If Y=3 Then
        A=Int(Rnd*4)
      Else
        A=Int(Rnd*3)+1
      End If
      Select Case A
        Case 0:XT=0 :YT=-1
        Case 1:XT=1 :YT=0
        Case 2:XT=0 :YT=1
        Case 3:XT=-1:YT=0
      End Select
    Wend
    M(X+XT,Y+YT)=1
  Next
Next
```

四方を壁で囲む

棒倒しアルゴリズムで迷路作成
〔教科書 P116～P117〕

Step02　作成した迷路をグラフィックで画面に表示する。

```
For Y=1 To 101 Step 1
  For X=1 To 101 Step 1
    If M(X,Y)=1 Then
      Line(X*5-5,Y*5-5)-(X*5,Y*5),1,BF
    End If
  Next
Next
Line(99*5,99*5)-(100*5,100*5),12,BF
```



Step03　カーソルキーで迷路を歩けるようにする。

画面右下の赤い部分(X=99,Y=99 の位置)に到達するとゴールです。

```
X=2:Y=2:ST=0
While(ST=0)
  Line(X*5-5,Y*5-5)-(X*5,Y*5),13,BF
  If X=99 And Y=99 Then
    Cls:Print "ゴール"
    End
  End If
  A$=""
  While(A$="")
    A$=Inkey$
  Wend
  BX=X:BY=Y
  Select Case ASC(A$)
    Case 27:ST=1
    Case 28:X=X+1
    Case 29:X=X-1
    Case 30:Y=Y-1
    Case 31:Y=Y+1
  End Select
  If M(X,Y)=1 Then
    X=BX:Y=BY
  End If
  Line(BX*5-5,BY*5-5)-(BX*5,BY*5),14,BF
Wend
```

座標(99,99)ならゴール

キー入力処理

押した方向によって X と Y の値を変更
ESC(27 番)を押したとき、ST の値を 1 にして、繰返しを抜けるようにする。

移動先が壁(ﾏｯﾌﾟﾃﾞｰﾀ 1)ならば、移動する前の X,Y のバックアップ BX,BY に戻す

キー入力の詳細や壁を貫通できないようにする処理については、ゲームプログラミング No.02 のプリントで解説しています。

問題　スタート位置とゴールの位置を変更してみよう。

水産海洋技術〔ゲームプログラミング〕　No.04

■ ブラックジャック(BlackJack)の作成

トランプを使った、ブラックジャック(BlackJack)の作成をします。
〔ルール〕
最初に２枚のカードを配ります。
カードの合計が 21 に近いほど強く、21 を超えると負けとなります。
カードの 10 以上は、10 として加算し、１は１または 11 として加算できます。

Step01　グラフィック画面の初期設定

```
Randomize
GBackColor=2:GForeColor=15
GScreen(600,100)
```

Step02　カードを作成します

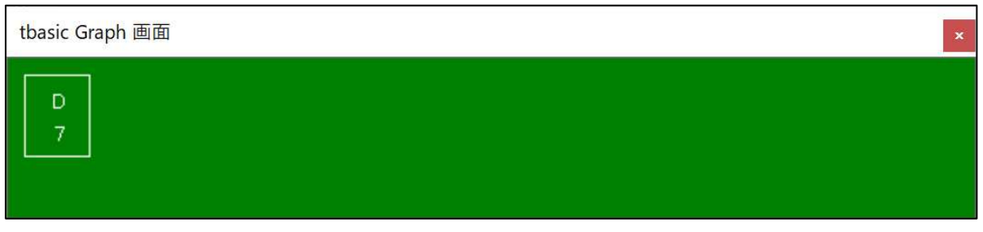
```
Dim T(52),M(52)                          ← 配列Ｔをカードの数字、Ｍをマークとします
For I=1 to 13                            ← カードの数字は１～13
  For J=1 to 4                           ← マークは４種類　1:ﾀﾞｲﾔ　2:ﾊｰﾄ　3:ｸﾛｰﾊﾞ　4:ｽﾍﾟｰﾄﾞ
    T((I-1)*4+J)=I                       ← 配列の１番目～52 番目となるように計算しています。
    M((I-1)*4+J)=J
  Next
Next
```

Step03　カードをシャッフルします

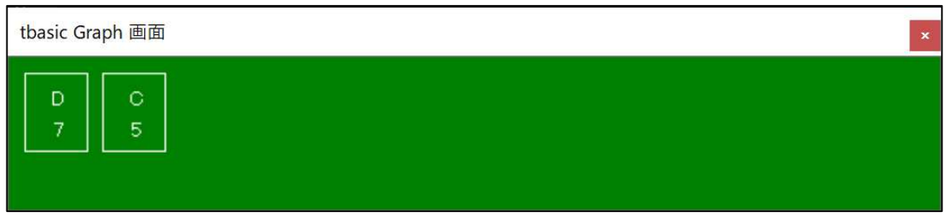
```
For I=1 to 100                           ← 100 回シャッフルします。
  A=Int(Rnd*52)+1                        ← 1～52 のランダムを２つ出して、
  B=Int(Rnd*52)+1                          その２枚を入れ替えます。
  C=T(A):T(A)=T(B):T(B)=C                ← カードの数字の入れ替え
  C=M(A):M(A)=M(B):M(B)=C                ← マークの入れ替え
Next
```

Step04　カードを表示します

```
Line(10,10)-(50,60),15,B                 ← 白い四角を描画
GLocate(28,20):GPrint Mid$("DHSC",M(1),1)← １枚目のマークを DHSC で表示
GLocate(25,40):GPrint T(1)               ← １枚目のカードの数字を表示
```

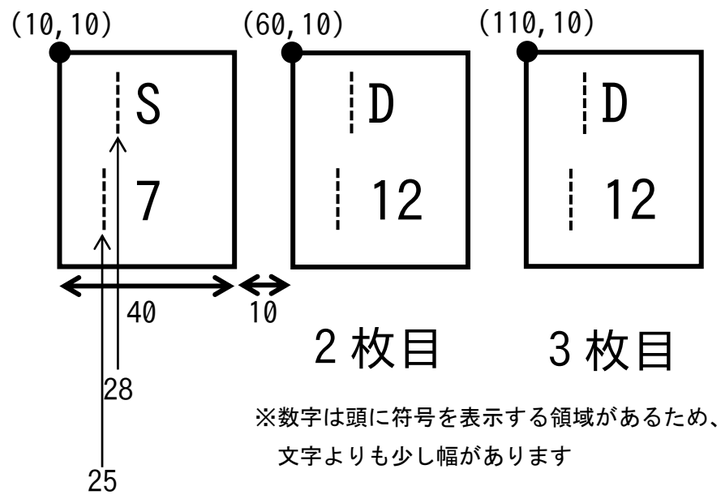


Step05　２枚目以降のカードを表示します



２枚目以降のカードは、右にずれて表示させたいため、計算をします。
何枚目のカードかを、変数Ｃを使って記憶します。

```
C=2
Line(C*50-40,10)-(C*50,60),15,B
GLocate(C*50-22,20):GPrint Mid$("DHSC",M(C),1)
GLocate(C*50-25,40):GPrint T(C)
```



※数字は頭に符号を表示する領域があるため、
文字よりも少し幅があります

Mid$(　)は、文字列の、指定番目から、指定文字数抜き出す命令。

（例）　A$=Mid$("ABCDEFG",2,3)　　← 文字列の２番目から３文字を A$に代入する

カードのマーク(配列 M)は、
１：ダイヤ(D)、２：ハート(H)、３：スペード(S)、４：クローバ(C)、
としているため、

```
Mid$("DHSC",M(C),1)
```

によって、マークの１～４を、文字の DHSC に変換している。

Step06　もう一枚引くかどうかをキー入力する


```
GLocate(100,80):GPrint ”もう一枚引きますか？［Y］はい［その他］いいえ”
K$=””
While(K$=””)
  K$=Inkey$
WEnd
C=C+1                                ← 次のカードとするため C を加算する。
```

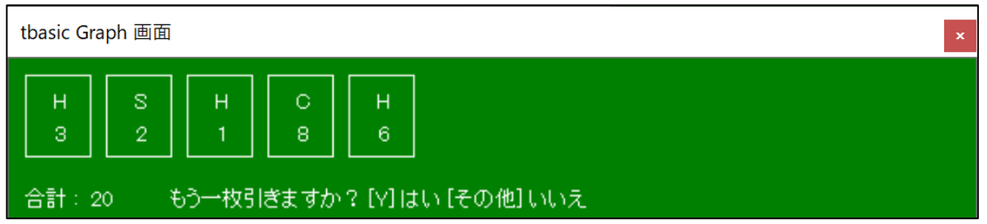
Step07　″Y″以外が押されるまで繰り返しとする

ここまでのプログラムの全体像

```
Randomize
GBackColor=2:GForeColor=15
GScreen(600,100)
Dim T(52),M(52)
For I=1 to 13
  For J=1 to 4
    T((I-1)*4+J)=I
    M((I-1)*4+J)=J
  Next
Next
For I=1 to 100
  A=Int(Rnd*52)+1
  B=Int(Rnd*52)+1
  C=T(A):T(A)=T(B):T(B)=C
  C=M(A):M(A)=M(B):M(B)=C
Next
Line(10,10)-(50,60),15,B
GLocate(28,20):GPrint Mid$(”DHSC”,M(1),1)
GLocate(25,40):GPrint T(1)
C=2
K$=”1”
While(K$=”1”)
  Line(C*50-40,10)-(C*50,60),15,B
  GLocate(C*50-22,20):GPrint Mid$(”DHSC”,M(C),1)
  GLocate(C*50-25,40):GPrint T(C)
  GLocate(100,80):GPrint ”もう一枚引きますか？［Y］はい［その他］いいえ”
  K$=””
  While(K$=””)
    K$=Inkey$
  WEnd
  C=C+1
WEnd
```

プログラム前半部分

Step08　合計値の計算と、21 を超えたときの負け判定



プログラム前半部分

```
C=2
K$=”1”
While(K$=”1”)
  Line(C*50-40,10)-(C*50,60),15,B
  GLocate(C*50-22,20):GPrint Mid$(”DHSC”,M(C),1)
  GLocate(C*50-25,40):GPrint T(C)
  SUM=0
  For I=1 To C
    If T(I)>=10 Then
      SUM=SUM+10
    ElseIf T(I)=1 Then
      SUM=SUM+11
    Else
      SUM=SUM+T(I)
    End If
  Next
  If SUM>21 Then
    SUM=0
    For I=1 To C
      If T(I)>=10 Then
        SUM=SUM+10
      Else
        SUM=SUM+T(I)
      End If
    Next
  End If
  Line(0,80)-(600,100),2,BF
  GLocate(10,80):GPrint ”合計 : ”+Str$(SUM)
  If SUM>21 Then
    GLocate(100,80):GPrint ”あなたの負けです。”
    End
  End If
  GLocate(100,80):GPrint ”もう一枚引きますか？［Y］はい［その他］いいえ”
  K$=””
  While(K$=””)
    K$=Inkey$
  WEnd
  C=C+1
WEnd
```

合計の計算
カードの数字 10 以上は 10 として加算
カードの 1 は、11 として加算、
それ以外は数字の値のまま加算

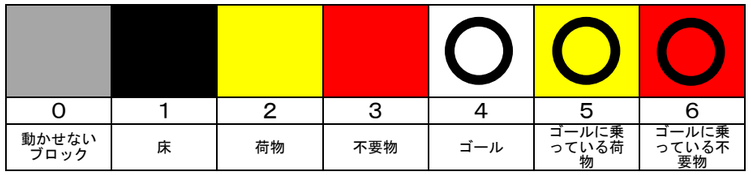
合計が 21 を超えていたら、
カードの数字 1 を 1 として計算し直す

水産海洋技術〔ゲームプログラミング〕　No.05

■ 倉庫番の作成

ゲーム「倉庫番」とは、すべての荷物をゴール地点まで運べばクリアとなるパズルゲームです。荷物の他に、外壁など押せないブロックと、荷物以外の押せるブロックが存在します。また、荷物は引くことができず、２個以上を押すこともできません。

今回は、以下のように荷物やブロックの値を設定する。



0	1	2	3	4	5	6
動かせないブロック	床	荷物	不要物	ゴール	ゴールに乗っている荷物	ゴールに乗っている不要物

Step01　パズルデータの設定

```
GBackColor=Black : GScreen(600,500)
Dim M(12,10)                              ← 横 12 マス、縦 10 マス
For I=1 To 10
  Read A$
  For J=1 To 12
    M(J,I)=Val(Mid$(A$,J,1))              ← Data から読み取って、１つずつ
  Next                                      ２次元配列に格納する。
Next
Data ”000000000000”
Data ”000001110000”
Data ”000111010000”
Data ”011102111000”
Data ”010243401000”
Data ”011240421100”
Data ”001044420100”
Data ”001022011100”
Data ”001111110000”
Data ”000000000000”
X=6:Y=9:Z=7                               ← 自分の位置 X,Y と荷物の総数 Z
```

Step02　画面表示を行うサブルーチンを作成

```
Sub Hyouzi(X,Y,M())                       ← 自分の位置とパズルデータを受け取る
  Cls 3
  For I=1 To 10
    For J=1 To 12
      Select Case M(J,I)                  ← 値によって色を変えて表示する
        Case 0 : Line(J*50-50,I*50-50)-(J*50,I*50),8,BF
        Case 2 : Line(J*50-50,I*50-50)-(J*50,I*50),14,BF
        Case 3 : Line(J*50-50,I*50-50)-(J*50,I*50),12,BF
        Case 4 : Circle(J*50-25,I*50-25),10,14
        Case 5 : Line(J*50-50,I*50-50)-(J*50,I*50),14,BF
        Case 6 : Line(J*50-50,I*50-50)-(J*50,I*50),12,BF
      End Select
    Next
  Next
  Circle(X*50-25,Y*50-25),23,10:Circle(X*50-25,Y*50-25),15,10,,,,F
End Sub
```

Step03　キー入力と移動処理

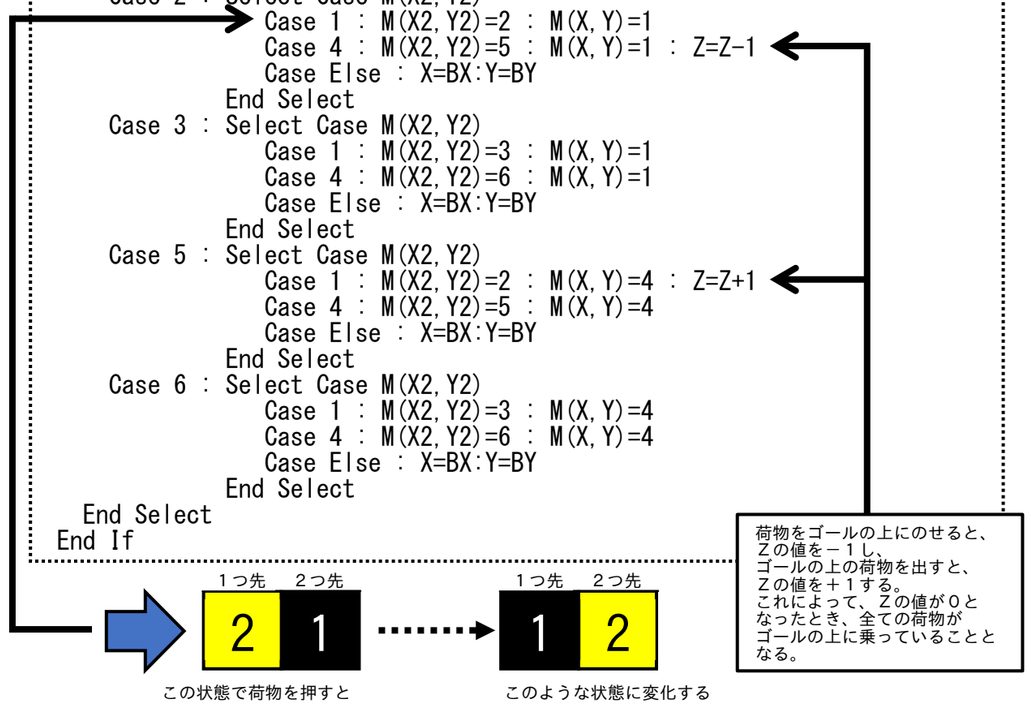
```
Call Hyouzi(X,Y,M())                      ← サブルーチンへ飛んで画面表示
A$=””
While(A$=””)
  A$=Inkey$
WEnd
BX=X:BY=Y
Select Case Asc(A$)
  Case 27 : End
  Case 28 : X=X+1:X2=X+1:Y2=Y             ← X,Y は移動先の座標へ変更
  Case 29 : X=X-1:X2=X-1:Y2=Y             ← X2,Y2 は２つ先の座標を代入
  Case 30 : Y=Y-1:X2=X:Y2=Y-1
  Case 31 : Y=Y+1:X2=X:Y2=Y+1
End Select
```

倉庫番では、荷物を押すときもう一つ先を見て押せるか押せないかを判断するため、1 つ先の座標と 2 つ先の座標を求めている。

Step04　荷物の移動処理

移動先（１つ先）が荷物か不要物で、その先（２つ先）が床かゴールなら押すことができる。
それ以外の組み合わせは、すべて押すことができない。

```
If M(X,Y)=0 Then                          ← 移動先が動かせないブロックのとき
  X=BX:Y=BY                               ← 移動前の値に戻す
Else
  Select Case M(X,Y)                      ← M(X,Y):1 つ先    M(X2,Y2):2 つ先
    Case 2 : Select Case M(X2,Y2)
               Case 1 : M(X2,Y2)=2 : M(X,Y)=1
               Case 4 : M(X2,Y2)=5 : M(X,Y)=1 : Z=Z-1
               Case Else : X=BX:Y=BY
             End Select
    Case 3 : Select Case M(X2,Y2)
               Case 1 : M(X2,Y2)=3 : M(X,Y)=1
               Case 4 : M(X2,Y2)=6 : M(X,Y)=1
               Case Else : X=BX:Y=BY
             End Select
    Case 5 : Select Case M(X2,Y2)
               Case 1 : M(X2,Y2)=2 : M(X,Y)=4 : Z=Z+1
               Case 4 : M(X2,Y2)=5 : M(X,Y)=4
               Case Else : X=BX:Y=BY
             End Select
    Case 6 : Select Case M(X2,Y2)
               Case 1 : M(X2,Y2)=3 : M(X,Y)=4
               Case 4 : M(X2,Y2)=6 : M(X,Y)=4
               Case Else : X=BX:Y=BY
             End Select
  End Select
End If
```



Step05　Ｚの値が０になるまで繰り返す

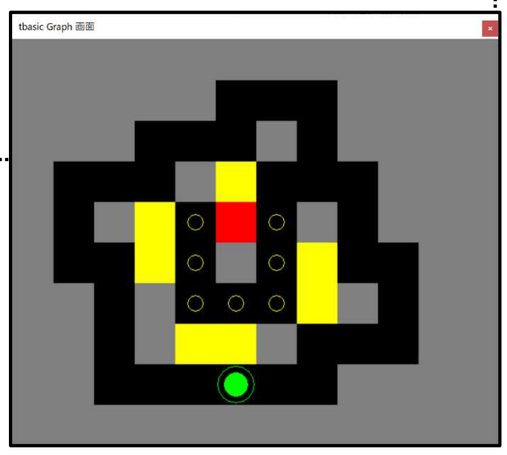
Step03～Step04 の処理を、Ｚの値が０となるまで繰り返す While 命令を入れる。
繰返し While 命令を抜けたとき、全ての荷物がゴールの上に乗っていることになる。

全体のプログラム

```
GBackColor=Black : GScreen(600,500)
Dim M(12,10)
For I=1 To 10
  Read A$
  For J=1 To 12
    M(J,I)=Val(Mid$(A$,J,1))
  Next
Next
Data ”000000000000”
Data ”000001110000”
Data ”000111010000”
Data ”011102111000”
Data ”010243401000”
Data ”011240421100”
Data ”001044420100”
Data ”001022011100”
Data ”001111110000”
Data ”000000000000”
X=6:Y=9:Z=7
```

```
Sub Hyouzi(X,Y,M())
  Cls 3
  For I=1 To 10
    For J=1 To 12
      Select Case M(J,I)
        Case 0 : Line(J*50-50,I*50-50)-(J*50,I*50),8,BF
        Case 2 : Line(J*50-50,I*50-50)-(J*50,I*50),14,BF
        Case 3 : Line(J*50-50,I*50-50)-(J*50,I*50),12,BF
        Case 4 : Circle(J*50-25,I*50-25),10,14
        Case 5 : Line(J*50-50,I*50-50)-(J*50,I*50),14,BF
        Case 6 : Line(J*50-50,I*50-50)-(J*50,I*50),12,BF
      End Select
    Next
  Next
  Circle(X*50-25,Y*50-25),23,10:Circle(X*50-25,Y*50-25),15,10,,,,F
End Sub
```

```
While(Z>0)
  Call Hyouzi(X,Y,M())
  A$=””
  While(A$=””)
    A$=Inkey$
  WEnd
  BX=X:BY=Y
  Select Case Asc(A$)
    Case 27 : End
    Case 28 : X=X+1:X2=X+1:Y2=Y
    Case 29 : X=X-1:X2=X-1:Y2=Y
    Case 30 : Y=Y-1:X2=X:Y2=Y-1
    Case 31 : Y=Y+1:X2=X:Y2=Y+1
  End Select
  If M(X,Y)=0 Then
    X=BX:Y=BY
  Else
    Select Case M(X,Y)
      Case 2 : Select Case M(X2,Y2)
                 Case 1 : M(X2,Y2)=2 : M(X,Y)=1
                 Case 4 : M(X2,Y2)=5 : M(X,Y)=1 : Z=Z-1
                 Case Else : X=BX:Y=BY
               End Select
      Case 3 : Select Case M(X2,Y2)
                 Case 1 : M(X2,Y2)=3 : M(X,Y)=1
                 Case 4 : M(X2,Y2)=6 : M(X,Y)=1
                 Case Else : X=BX:Y=BY
               End Select
      Case 5 : Select Case M(X2,Y2)
                 Case 1 : M(X2,Y2)=2 : M(X,Y)=4 : Z=Z+1
                 Case 4 : M(X2,Y2)=5 : M(X,Y)=4
                 Case Else : X=BX:Y=BY
               End Select
      Case 6 : Select Case M(X2,Y2)
                 Case 1 : M(X2,Y2)=3 : M(X,Y)=4
                 Case 4 : M(X2,Y2)=6 : M(X,Y)=4
                 Case Else : X=BX:Y=BY
               End Select
    End Select
  End If
WEnd
Call Hyouzi(X,Y,M())
Print ”ゲームクリアです。”
End
```



完成したら、インターネットで倉庫番のパズル面をしらべて、入力してみよう。