

練習問題

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

```
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

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。

問題2 上のプログラムを修正して、何かの計算を行うプログラムを作成してみよう。

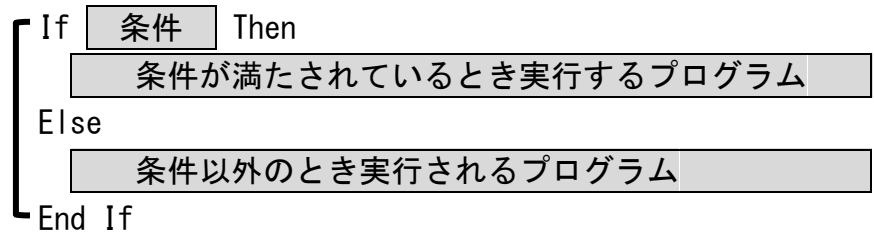
Step01 前回の確認

```
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 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

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。

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。

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。

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。

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。

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

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

問題 2 1 ~ 30 までの数値を表示しなさい。

問題 3 1 ~ 100 までの奇数を表示しなさい。

問題 4 1 ~ 100 までの偶数を表示しなさい。

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。

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 $R = \text{Int}(\text{Rnd} * 10) + 1$ 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出るので、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

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。

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。

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。

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ウが表示される。
 ※ウィンドウサイズは 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。

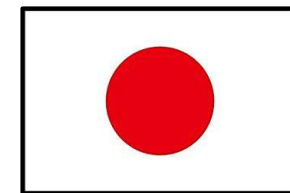
Circle(横座標, 縦座標), 半径, 色番号, 開始角度, 終了角度, 比率, [F]

```
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。

練習問題

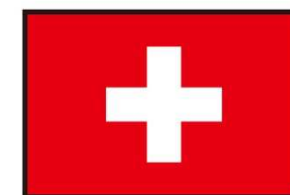
問題 1 日本の国旗を描いてみよう。



問題 2 フランス国旗を描いてみよう。



応用 1 スイス国旗を描いてみよう。

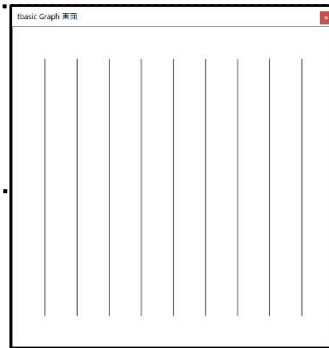


Step01 前回の確認

```
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
```

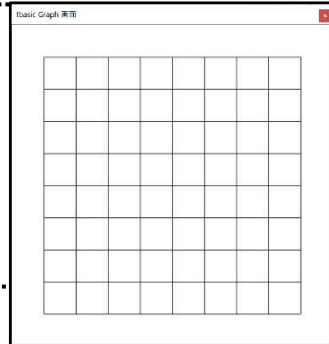


※横軸を 50 から 450 まで 50 間隔でずらして描いている。
 ※Step の間隔を変えることで、線の幅が変わります。

練習問題

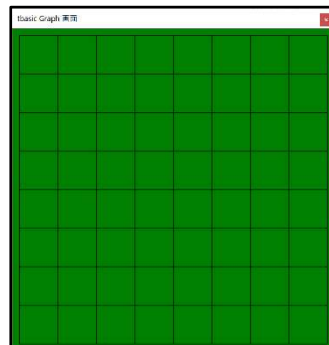
問題 1 続けて横線を描いてマス目に見よう。

```
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
```

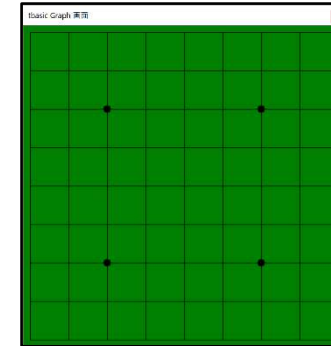


練習問題

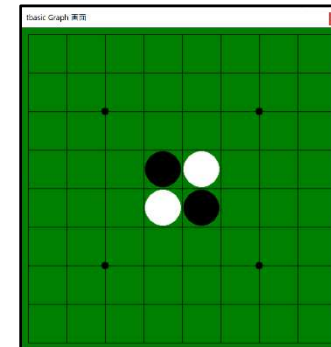
問題 1 背景を色番号 2 (緑) にし、黒色の線 (色番号 0) で 8 x 8 のオセロ盤を描いてみよう。ただし、四隅の余白は 10 ドットとする。



問題 2 オセロ盤の中央 4 x 4 マスの四隅に、黒い丸を描いてみよう。



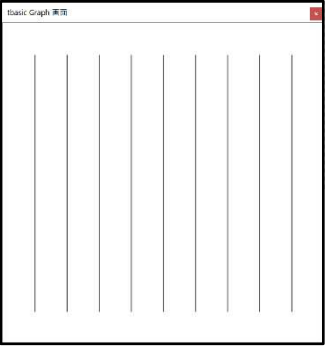
問題 3 オセロ盤に最初の駒を描いてみよう。



応用 1 オセロゲームを進めてみよう。

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
```

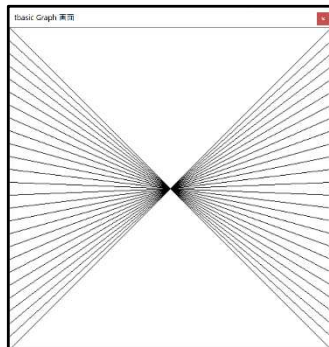


練習問題

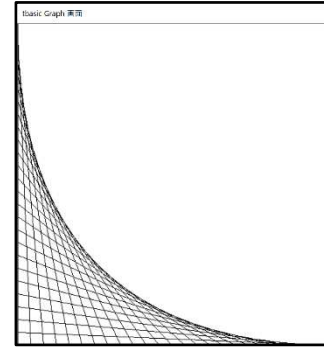
問題 1 次のような線を、繰り返し命令 For を使って描きなさい。
ただし、線の間隔は 20 ドットとする。



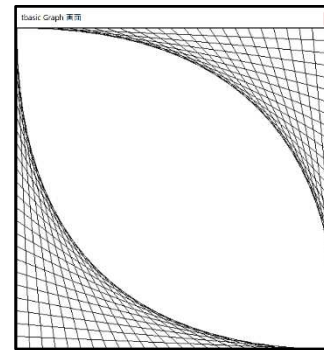
問題 2 次のような線を、繰り返し命令 For を使って描きなさい。



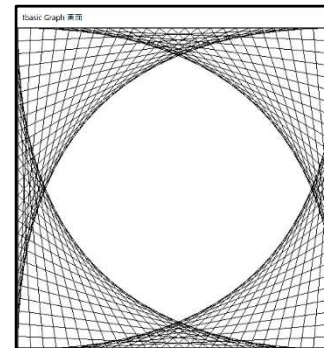
問題 3 次のような線を、繰り返し命令 For を使って描きなさい。



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。

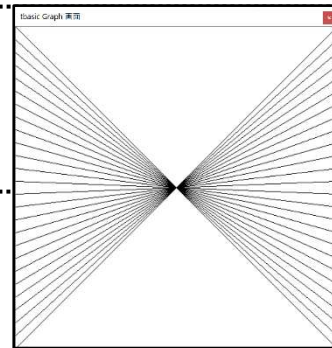


応用 2 次のような線を、繰り返し命令 For を使って描きなさい。



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。

```
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、読み込んだ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

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。

```
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。

```
Data 50,30,75,90,27,10,100,60,49,84,61,18,50,97,77
```


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

問題 1 8 個の数値を配列 A に読み込みなさい。

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。

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。

応用 2 読み込んだ数値を逆順にすべて表示しなさい。

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。

練習問題

問題1 8×8のキャラクターにしてみよう。

問題2 キャラクターを自由に変えてみよう。

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
```

※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

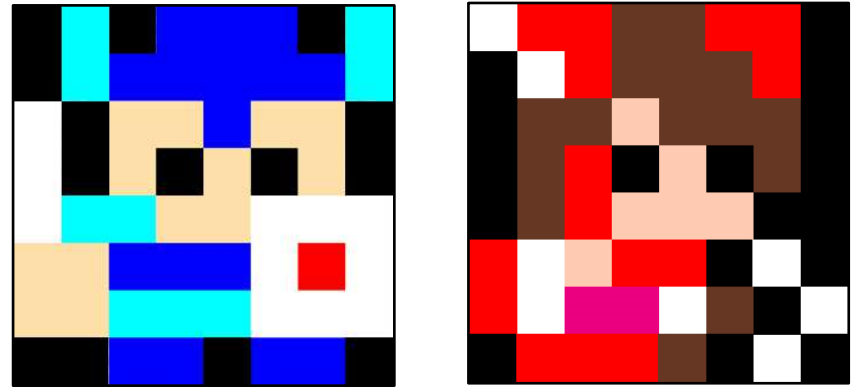
```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    Line((X-1)*20, (Y-1)*20)-(X*20, Y*20), C, BF
  Next
Next
Data 15,15,12,12,12,12,12,12
Data 15, 9, 6, 9, 6,14,14,15
Data 6,15, 6, 6, 6, 6, 6,15
Data 8,15, 6,14, 6, 6,15, 6
Data 8,12,12,12,12,12,12, 8
Data 15,15,12,12,12,12,15,15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

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。



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 "FFCCCCC"
Data "F9696EEF"
Data "6F66666F"
Data "8F6E66F6"
Data "8CCCCC8"
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。」程度の説明が良い。

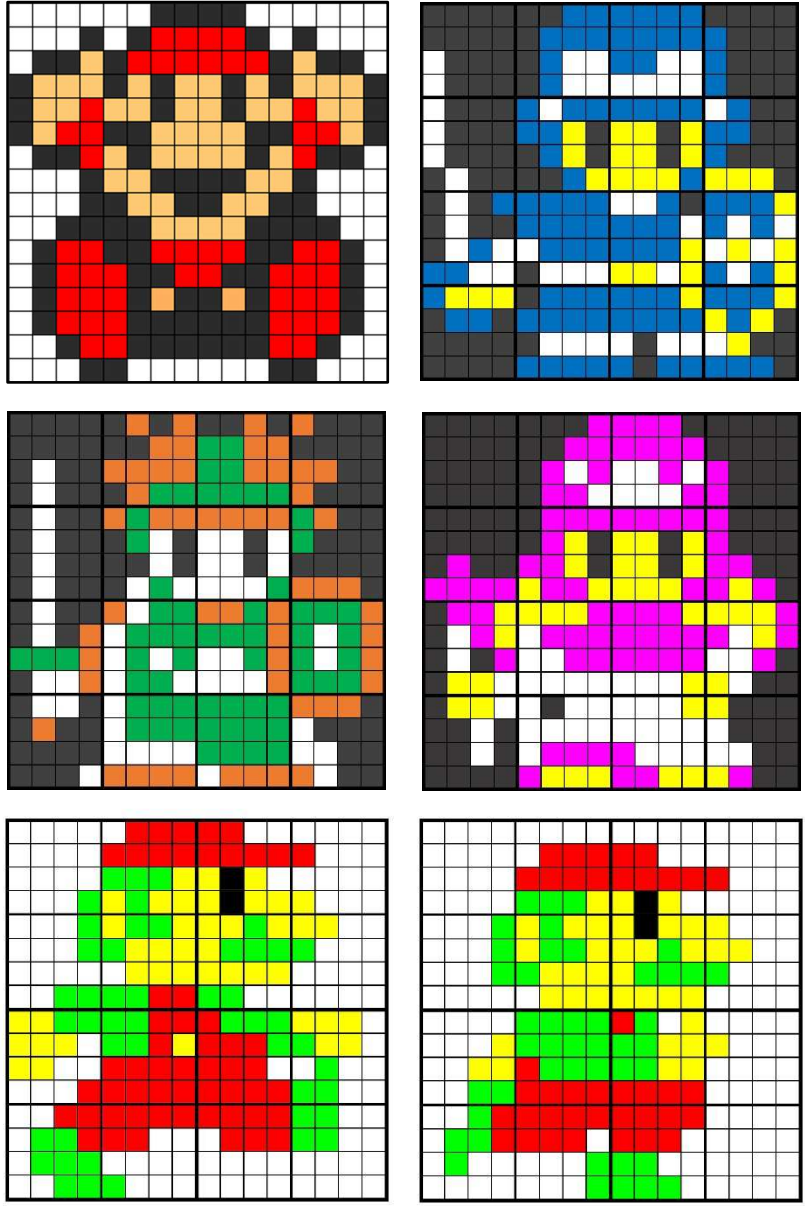
練習問題

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 "FFFFFFFFFFFFFF"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "F0000000000000F"
Data "FFFFFFFFFFFFFF"
    
```

問題2 好きなキャラクターを描いてみよう。



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() ← Call 命令で、サブルーチン呼び出す
End
```

```
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

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。

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。

```
Sub SIKAKU()
```

```
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。

Step01 前回の確認

```
GScreen(500,500)
Cls 2
Randomize
Call MARU() ← Call 命令で、サブルーチン呼び出す
End

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) ← カッコ内に渡す値を記述する
End

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

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

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

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)

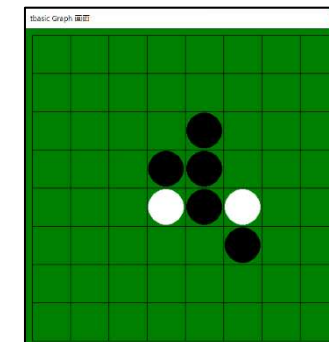
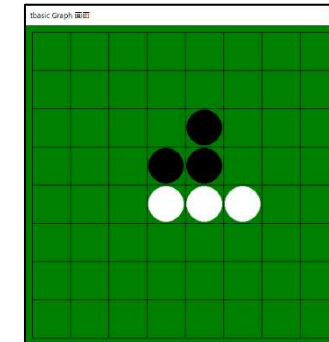
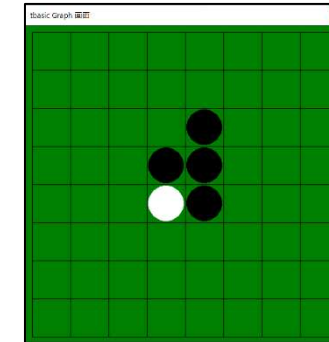
Call KOMA(5,3,0)
Call KOMA(5,4,0)

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

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

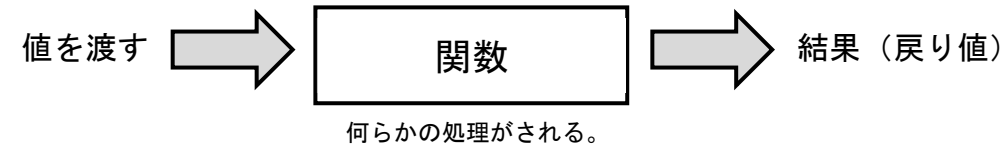
End

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



■ 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)
Print A
End
    
```

※テストの点数は 29 点

```

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

※テストの点数を変数 TEN で受け取る。

※返す値を設定する

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
    
```

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。

```

Function TESUTO(TEN)
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。

```

Function TESUTO(TEN)
End Function
    
```

■ 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
```

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
```

表示するプログラム



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。