

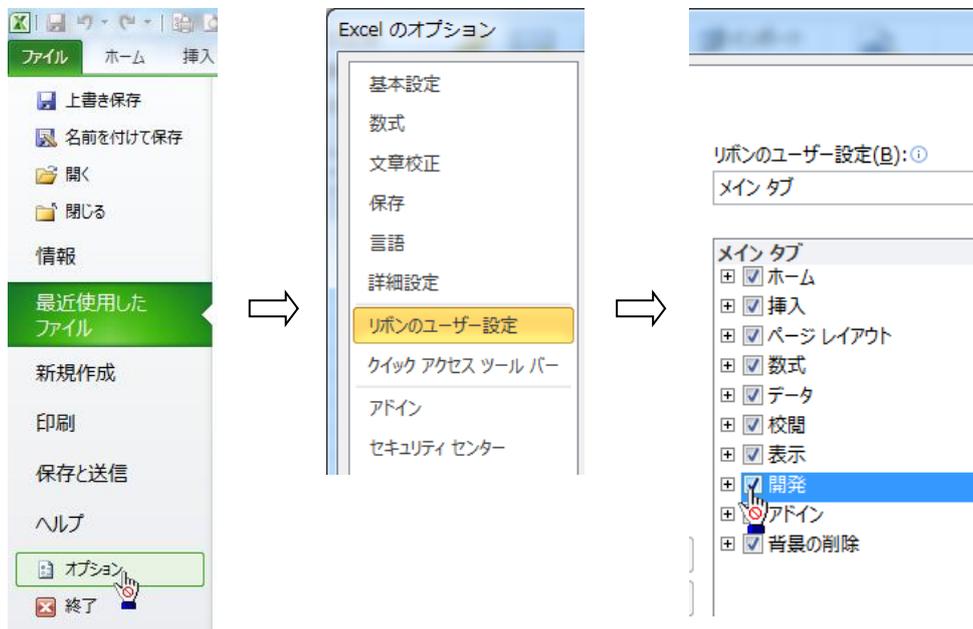
プログラム ～VBAによる自動処理～

VBAとは、『Visual Basic for Applications』の略で、簡単に言えばプログラミング言語の1つである。VBAを使ってプログラムを作ってみよう。

(1) Excelブックを開く。

以下の手順で、リボンに「開発」タブを表示させる。

- ① 「ファイル」タブから、「オプション」を選択する。
- ② 「Excelのオプション」ウィンドウで、左側のメニューから「リボンのユーザー設定」を選択する。
- ③ 「開発」タブをリボンに表示する」をクリックする。



- (2) 「開発」タブ→「挿入」ボタン→「ボタン」 (右図 I) をクリックし、ボタンを作成（四角の対角線をドラック 図 II）して、「マクロの登録」ダイアログで、「新規作成」ボタンをクリックすると、Visual Basic Editor が起動する。(図 III)

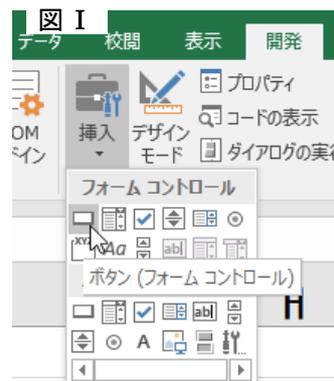


図 II

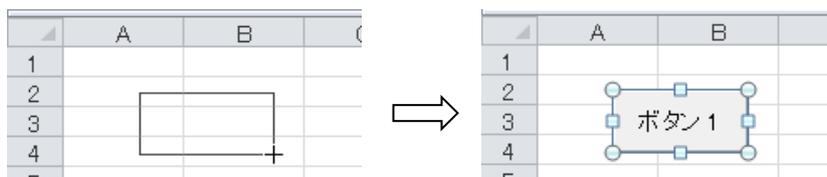


図 III



(3) 「Sub ボタン1_Click()」と「End Sub」の間に、プログラムを記述する。

① セル A8 に「3」を入力してみよう。

「Sub ボタン1_Click()」と「End Sub」の間の部分を (※) と表すことにする。(右図IV)

図 IV

```
Sub ボタン1_Click()
    (※)
End Sub
```

(※) の部分に、「Cells(8, 1) = 3」と記述する。(右図 V)

「Cells(8, 1)」は、8 行 1 列目のセルを指し、このセルに「3」を代入するという命令である。

見やすくするために、「Tab」キーを使って字下げして記述することにしよう。

タスクバーにある Excel のアイコンから、シートを選択して、「ボタン 1」をクリックすると、マクロが実行される。

図 V

```
Sub ボタン1_Click()
    Cells(8, 1) = 3
End Sub
```

② 「3+5」を計算して、その結果をセル A9 に表示させるには、(※)の部分に、

「Cells(9, 1) = 3+5」

と記述して、「ボタン 1」をクリックしてマクロを実行すればよい。

③ For …… Next ~ の利用

セル範囲 B8:B17 に「11~20」の連番を入力してみよう。

①のような命令を 10 行分「Cells(8, 2) = 11」……「Cells(17, 2) = 20」と記述してもできるが、繰り返しの操作は「For …… Next ~」を用いる。

(※) の部分に、右のように記述する。

入力ができたら、「ボタン 1」をクリックして、マクロを実行してみよう。

```
Sub ボタン1_Click()
    For i = 1 To 10
        Cells(7 + i, 2) = 10 + i
    Next i
End Sub
```

[解説]

- ❶ 1 行目の「For i = 1 To 10」は、「変数 i に 1 から順に 10 までの値を代入して、それ以降 Next までの処理を繰り返せ」という命令
- ❷ 繰り返し実行される処理が「Cells(7 + i, 2) = 10 + i」であり、ここの i に 1~10 を代入すると、セル範囲 B8:B17 に 11~20 の連番を表示させることになる。
- ❸ 「Next i」は、「i の値を 1 増やして処理を繰り返す」という意味。

- ④ B列に入力されたデータを削除してから、右図のように、最初の行の「For i = 1 To 10」のあとに、「Step 2」を付けて実行して、Step命令の意味を確かめてみよう。

```
Sub ボタン1_Click()
  For i = 1 To 10 Step 2
    Cells(7 + i, 2) = 10 + i
  Next i
End Sub
```

※ 実行する前に、セル範囲 B8:B17 の内容を削除しましょう。

① 最大公約数を求めるプログラム ～ユークリッドの互除法を活用～

- ユークリッドの互除法によって、最大公約数を求めるプログラムを記述してみよう。ユークリッドの互除法の原理は、次の定理である。
 a, b の最大公約数を $\text{gcd}(a, b)$ で表すことにすれば、次の定理が成立する。

定理 a, b を正の整数とする。

$$a = bq + r \quad (0 < r < b) \quad \text{のとき,} \quad \text{gcd}(a, b) = \text{gcd}(b, r)$$

この定理を利用して、884 と 403 の最大公約数を求めてみよう。

884 を 403 で割ると商が 2、余り 78 ($884 = 403 \times 2 + 78$)

$$\text{gcd}(884, 403) = \text{gcd}(403, 78)$$

403 を 78 で割ると商が 5、余り 13 ($403 = 78 \times 5 + 13$)

$$\text{gcd}(403, 78) = \text{gcd}(78, 13)$$

78 を 13 で割ると商が 6、余り 0 ($78 = 13 \times 6$)

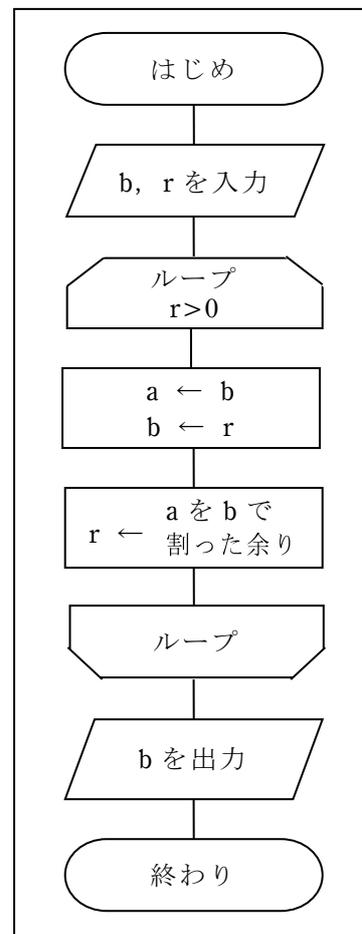
$$\text{gcd}(78, 13) = 13$$

- ① G.C.D. (最大公約数) を求める手順を確認すると、次のようになる。

[手順]

- ① 最大公約数を求める 2 数を入力し、それを仮に b, r とする。
- ② a に b を代入し、 b に r を代入する。
 a を b で割って余り r を求める。
- ③ $r = 0$ のとき、 b が G.C.D. である。
 $r > 0$ のとき、処理②に戻って処理を繰り返す。

これを流れ図 (フローチャート) で表すと、右のようになる。



② 最大公約数を求めるプログラム（マクロ）をボタンに登録する

まず登録するボタンを新規に作成する。

- ・「開発」タブ→「挿入」ボタン→「ボタン」 をクリックして、四角形の対角線をドラッグしてボタンを作成する。「マクロの登録」ダイアログで、「新規作成」ボタンをクリックすると、Visual Basic Editor が起動する。
- ・次のプログラムは、「Sub ボタン2_Click()」と「End Sub」の間の部分に記述する。この部分を(※2) とする。

③ セル H2, H3 に入力された値を a, b として、その G.C.D. (最大公約数) を求めるプログラムを作ってみよう。

①の手順（流れ図）に従ったプログラムを(※2) の部分に次のように記述する。

判断◇の部分は、分岐処理を行うために、

While (条件) Wend

を利用する。条件を満たす間、処理が繰り返される。

④ セル H2, H3 に値を入力して、「ボタン 2」をクリックして最大公約数を求めよう。

```
Sub ボタン2_Click()  
    b = Cells(2, 8)  
    r = Cells(3, 8)  
    While r > 0  
        a = b  
        b = r  
        r = a Mod b  
    Wend  
    Cells(5, 8) = b  
End Sub
```

[プログラムの解説]

Sub ボタン2_Click()	マクロ名「ボタン2_Click」
b = Cells(2, 8)	セル H2 の値を変数 b に代入する
r = Cells(3, 8)	セル H3 の値を変数 r に代入する
While r > 0	変数 r が条件 r > 0 を満たす間、処理が繰り返される
a = b	変数 b の値を変数 a に代入する
b = r	変数 r の値を変数 b に代入する
r = a Mod b	変数 a を変数 b で割ったときの余りを変数 r に代入する
Wend	変数 r の値が r > 0 ならば、処理を while に移す
Cells(5, 8) = b	セル H5 に変数 b の値に代入する（変数 b の値に表示）
End Sub	マクロはここまで

素数判定プログラム

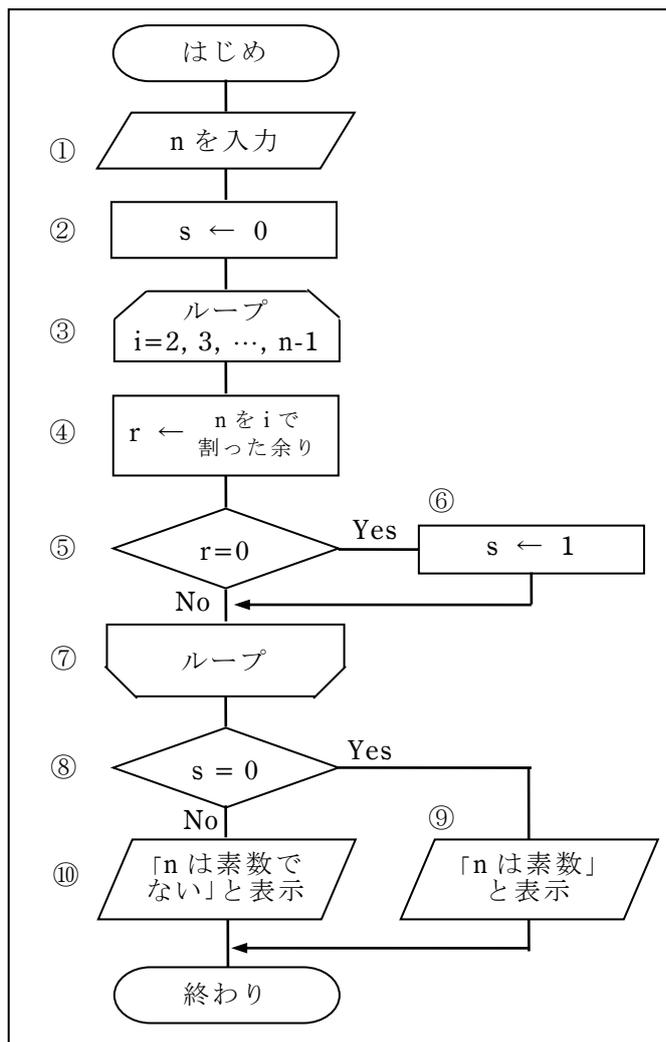
与えられた整数が、素数であるかどうかを判定するプログラムを作ってみよう。

与えられた整数より小さい正の整数すべてで割ってみて割り切れるかどうかを調べることにより判定する方法である。その手順は、次のように表すことができる。

[手順]

- ① 判定する 2 以上の整数を入力し、それを n とする。
- ② 初期値として s の値を 0 とする。
- ③ i=2~n-1 に対して、n を i で割った余り r を求める。
- ④ r=0 のとき、s の値を 1 にする。
r>0 のとき、そのまま。あとは③を繰り返す。
- ⑤ s=0 のとき、n は素数、そうでないとき n は素数でない

この手順を流れ図（フローチャート）で表すと，次のようになる。



上のアルゴリズムに従って素数判定のプログラムを作ると，以下のようになる。

```
Sub 素数判定()
    n = Cells(1, 2)
    s = 0
    For i = 2 To Int(Sqr(n))
        r = n Mod i
        If r = 0 Then
            s = 1
        End If
    Next i
    If s = 0 Then
        Cells(2, 2) = "nは素数である"
    Else
        Cells(2, 2) = "nは素数でない"
    End If
End Sub
```

1 エラトステネスのふるい

配列を利用して素数の表を作成しよう。

(1) 2以上の整数を順に書き並べた表を考える。まず、2より大きい2の倍数を取り除く。次に、3(=2+1)より大きい3の倍数を取り除く。次は4(=3+1)であるが、4より大きい4の倍数はすでに除かれている。しかし、ここではこれも繰り返す。次に5(=4+1)より大きい5の倍数を取り除く。この手順を繰り返していくと、最後に素数が残る。

(2) この考え方で、セルC1に入力された整数以下の素数を求めるプログラムを作ると、右のようになる。

素数は、A列の1行目から記入されるようになっていく。

(3) マクロの実行

作成したボタンをクリックして実行する。

【解説】「Dim s(1000000)」は、「s」という名前の10000001個の変数の配列s(0), s(1), …, s(1000000)を用意したことを表している。

```
Sub sosu()  
    Dim s(1000000)  
    m = Cells(1, 3)  
    For i = 2 To m  
        s(i) = 0  
    Next i  
    For i = 2 To m  
        For j = i * i To m Step i  
            s(j) = 1  
        Next j  
    Next i  
    k = 1  
    For i = 2 To m  
        If s(i) = 0 Then  
            Cells(k, 1) = i  
            k = k + 1  
        End If  
    Next i  
End Sub
```