Excel VBAを用いたプログラミング　第1部

奈良教育大学　　薮 哲郎

最終更新　　2017.6.5

# はじめに

　プログラミングはコンピューターの最も効果的な使い方です。プログラムを組んでコンピューターに実行させると、人間が一生かかっても出来ない量の計算を、コンピューターは瞬時に終わらせます。本演習ではプログラミング環境としてExcel VBAを使用します。Excelは成績処理、住所録の管理などに必須のソフトウェアであり、Excel VBAでプログラムを組めるようになることは、あなた方の将来の仕事を大いに助けることになるでしょう。

　Excel VBAについての何らかの参考書を購入することをお薦めします。ある程度Excel VBAに慣れたなら、「Excel VBA パーフェクトマスター」か「できる大辞典　Excel VBA」のいずれか一冊を常備しておくと良いでしょう。入門者向きの書籍としては「かんたんプログラミング　Excel VBA　基礎編　大村あつし」が分かりやすいでしょう。

# プログラミングの準備

注意

　本演習はExcel2013を用いて行います。セーブするときは必ず「Excelマクロ有効ブック（拡張子xlsm）」を使って下さい。「Excelブック（拡張子xlsx）」ではマクロがセーブされないので、せっかく作ったプログラムが失われてしまいます。

　「ファイル」→「オプション」→「リボンのユーザー設定」で「開発」にチェックを入れて下さい。「開発」タブが出現します。

## Excelの概念

　ファイル「○○.xlsm」は次のような概念構造を持っています。

* 1つのxlsmファイルは1つのプロジェクトを表し、ワークシート以外に、標準モジュールなどを含むことが出来る。
* VBAエディタを使うとxlsmファイルの内部構造を見ることが出来る。
* プログラムは標準モジュールの中に書く。

　「開発」タブの「コード：Visual Basic」をクリックすると、Visual Basic Editorが起動し、プログラムを編集できる状態になります。

　左側の「プロジェクト」と書いてあるフレームに、xlsmファイルの概念構造を表すツリー図が描かれています。「プロジェクト」フレームがない場合は、Visual Basic Editorの「表示」→「プロジェクト エクスプローラ」でこのフレームを表示して下さい。

## マクロ

　エクセルにおける操作（セルに数値を入れる、セルの内容をコピーするなど）はマクロとして記録することが出来ます。「開発」→「コード：マクロの記録」をクリックして「OK」をクリックすると、マクロの記録を開始します。「■ 記録終了」を押すと、マクロの記録を終了します。

　「開発」→「Visual Basic」でVisual Basic EditorをActiveにすると、標準モジュールの中にModule1という名前のモジュールが新規生成され、その中に記録したマクロが格納されています。「Module1」をダブルクリックすると内容が表示されます。マクロは

Sub マクロ名()

End Sub

という形式になっています。Excel VBAにおいては「マクロ ＝ プログラム」です。

　マクロを実行するには以下のように操作します。

1. マクロの中にカーソルを置く、あるいは「プロシージャ」と表示されるリストボックスで実行したいマクロ名を選ぶ。
2. 「実行」→「Sub／ユーザーフォームの実行」あるいはツールバー中の  をクリックする。

## プログラム作成のための基本操作

* VBAProjectの下に標準モジュールが存在しない場合は「挿入」→「標準モジュール」でプログラムを記述するためのモジュールを作成します。
* モジュールを削除するには「ファイル」→「Module xの解放」と操作します。
* モジュールの切替はモジュール名をダブルクリックです。シングルクリックでは何も起こりません。
* モジュールの名前を変更するには「表示」→「プロパティウィンドウ」でプロパティのフレームを表示し、オブジェクト名を変更します。
* マクロ名として使える文字は「ローマ字」「数字」「アンダースコア ( \_ )」です。アンダースコアを使うと「calc\_sum\_value()」のように分かりやすいマクロ名を付けることが出来ます。ただし、先頭の文字として数字は使えません。
* マクロの実行を中断するには「Esc」キーを押します。無限ループに陥ったときなどに押します。
* マクロ実行時にエラーが発生したとき「デバッグ(D)」を選択すると、エラーが発生した行にカーソルが移り、黄色矢印が表示されます。実行は中断されています。イミディエイトウィンドウで変数の内容を表示させることが出来ます。「実行」→「リセット」により、実行を終了します。
* イミディエイトウィンドウを表示させるには「表示」→「イミディエイトウィンドウ」と操作します。
* 誤ってF9キーを押してブレークポイントを設定してしまったときは（設定した行が茶色くなり、行の左に茶色の丸印が付きます）、もう一度F9キーを押します。あるいは「デバッグ」→「全てのブレークポイントの解除」と操作します。

## ヘルプ

　Visual Basic Editorのコードウィンドウ上で調べたい命令語（例えば For）の上にカーソルを置き、F1キーを押すと、その命令語に対するヘルプが開きます。

# Forループと変数の使用

　まずは、簡単なプログラムを作ってみましょう。

＜例題1＞

　セルA1に文字列abcを入れ、セルA2に数値60を入れるプログラムを作成しなさい。

学ぶ命令

　Sub, End Sub, Range, Value, Cells, コメント文

Sub rei1()

 Range("A1").Value = "abc" ' 文字列は " で囲む ' 以下はコメント

 Range("A2").Value = 60

End Sub

　文字列は "abc" のようにダブルクォーテーションで囲みます。Range命令の代わりにCells命令を使うと以下のようになります。

Sub rei1b()

 Cells(1, 1).Value = "abc" ' Cells(行番号,列番号)

 Cells(2, 1).Value = 60

End Sub

あるいは

Sub rei1c()

 Cells(1, "A").Value = "abc" ' Cells(行番号,列名)

 Cells(2, "A").Value = 60

End Sub

　.Valueは省略できるので、以後は省略します。

＜例題2＞

　以下のプログラムを実行し、文字列と数値の違いを確認しなさい。

Sub rei2()

 Range("A1") = "abc"

 Range("A2") = abc

 Range("A3") = "10" + "20"

 Range("A4") = 10 + 20

End Sub

＜例題3＞

　A1～A5までの全てのセルに数値10を入れるプログラムを作成しなさい。

学ぶ命令

　変数, For, CStr

　普通に考えると次のようなプログラムが考えられます。

Sub rei3()

 Cells(1, 1) = 10

 Cells(2, 1) = 10

 Cells(3, 1) = 10

 Cells(4, 1) = 10

 Cells(5, 1) = 10

End Sub

　1, 2, 3, 4, 5 と変化している部分を変数 i で表し、i の値を1, 2, 3, 4, 5 と変化させると、プログラムが簡潔に書けます。また、このようにプログラムを書くと、A1～A100までのセルに値を代入する場合にも対応できます。

Sub rei3b()

 For i = 1 To 5

 Cells(i, 1) = 10

 Next i

End Sub

　Range命令を使う場合は、Range("A1") = 10, Range("A2") = 10, ..... のように、Forループの中で文字列を "A1", "A2", "A3",.... のように変化させる必要があります。以下のように書きます。

Range("A" + CStr(i)) = 10

　CStr(i) は数値iを文字列に変換する命令です。例えば i = 2 のとき、CStr(i) は文字列 "2" を表します。+ は文字列を連結する命令で、例えば "A" + "2" = "A2" となります。文字列の連結は + 以外に & を使うこともできます。& を使う場合、次のようにCStrを省略することができます。

Range("A" & i) = 10

Debug.Print でイミディエイトウィンドウに書き出す

　VBAエディタで「表示」→「イミディエイトウィンドウ」と操作すると、イミディエイトウィンドウが出現します。

　Debug.Printという命令で、イミディエイトウィンドウに変数の内容などを表示することが出来ます。For文を理解するために、以下のように打ってみましょう。iの値を更新しながら、For ～ Next の間を5回繰り返すことが分かります。

Sub rei3c()

 Debug.Print "---- " & Now() & " ----"

 For i = 1 To 5

 Debug.Print "i =" & i

 Next i

End Sub

　Debug.Print の後に記述した文字列、数値はイミディエイトウィンドウに表示されます。Now() は関数呼び出し命令で、現在の日付と時刻に置き換わります。「&」は数値や文字列を連結する記号です。「&」で数値や文字列を連結する場合、数値は自動的に文字列に変換されます。「&」の代わりに「+」を使うこともできますが、「+」は文字列を連結することしかできません。数値を扱う場合、"a = " + CStr(i) + " [V]" のように文字列に変換する必要があります。

　数値や文字列を連結する記号は「&」と「+」以外に「;」「,」があります。

　「;」は正の数値の手前に半角スペースが1個入ります。

　「,」は次の出力位置（1, 15, 30, 45...文字目）まで送ります。

＜例題4＞

　まず、A1のセルに任意の数値（例えば1）を手作業で入れる。

　この状態で、A2以下のセルを、A2 = A1 \* 2, A3 = A2 \* 2, ..... のように、一つ上のセルの値を2倍し、A2～A10までのセルを埋めるプログラムを書きなさい。

Sub nibai()

 For i = 2 To 10

 Cells(i, 1) = Cells(i - 1, 1) \* 2

 Next i

End Sub

　セルA2に「= A1 \* 2」という数式を入れ、A3以下のセルにドラッグしても同じことはできます。しかし、「\* 3」に変更したいとき、プログラムは1箇所書き換えるだけで済みますが、セル内の数式を使うと、全てのセルの数式を書き換える必要があります。

エラーで停止したときのイミディエイトウィンドウ

　上記のプログラムを　For i = 1 To 10　とすると、エラーが発生してプログラムが止まってしまいます。エラーを知らせるウィンドウの「デバッグ」をクリックすると、黄色矢印でエラーが発生した箇所を示します。エラーが発生した状態で停止しています。イミディエイトウィンドウで

Print i　　　あるいは　　　? i

と打つと、エラーが発生してプログラムが停止した時点のiの値を見ることができます。i = 1 なので、Cells(0, 1) に値を入れようとしたが、行番号0のセルはない（1から始まる）ので、エラーが発生していることが分かります。

＜課題1＞

　A1とA2に1を入れる。An = An1 + An2 で表されるフィボナッチ数列をA3～A10に作成するプログラムを書きなさい。プログラム名を　Sub kadai\_1()　として作成しなさい。以下、課題xに対する答のプログラムは　Sub kadai\_x()　としなさい。

＜例題5＞

　1から10までの和を求めてA1に書き込むプログラムを作成しなさい。

　和の途中経過を書き込む変数 sum を導入し、以下のように考えます。

Sub goukei()

 sum = 0

 sum = sum + 1

 sum = sum + 2

 sum = sum + 3

 sum = sum + 4

 ........

 sum = sum + 10

 Range("A1") = sum

End Sub

　ここで sum = sum + 1 のような部分があります。コンピューターのプログラムでは「=」は右辺の値を左辺に代入するという意味を持っています。1, 2, 3, .... 10 と変化する部分をFor ～ Next を用いて書き直すと以下のようになります。

Sub goukei()

 sum = 0

 For i = 1 To 10

 sum = sum + i

 Next i

 Range("A1") = sum

End Sub

　このプログラムの中に

sum = sum + i

という部分があります。私はこの1行がプログラミングの神髄を表していると思います。変数sumの値を更新しながら処理を進めるという概念をマスターして下さい。

　なお、最初にsum = 0 として変数sumを初期化しています。VBAの場合、変数の初期値は、数値として扱う場合は0、文字列として扱う場合は ""（長さ0の文字列）となります。従って、sum = 0 を省略しても動作しますが、C言語のように変数の初期値が不定である処理系もあるので、初期化をする習慣をつけて下さい。

　Excel VBAではSubの後に書くプロシジャー名と同名の変数名を使うことはできません。この例ではプロシジャー名をgoukei, 変数名をsumとしています。

＜課題2＞

　7の階乗を求めてA1に記入するプログラムを作成しなさい。

------------------------------------------------------

ここまでのまとめ

1. セルへの代入方法　　Range と Cells の2つの方法

2. 文字と数値の違い　　12 + 3 と "12" + "3" の違い

3. Forの使い方

4. i = i + 1あるいは i = i \* 2 というパターンの使い方

------------------------------------------------------------------------

＜例題6＞

　1, 2, 4, 8, ‥‥‥ という数列の第10項までをA1, A2, A3・・・A10に入れ、その合計をB1に記入するプログラムを作成しなさい。

Sub suuretu\_no\_wa ()

 k = 1

 sum = 0

 For i = 1 To 10

 Range("A" & i) = k

 sum = sum + k

 k = k \* 2

 Next i

 Range("B1") = sum

End Sub

　以下の例題では VBA1.xlsm を使います。

＜例題7＞

　シート「成績処理」をActiveにして下さい。D2～D11に英語、E2～E11に数学の成績が入力されています。F2～F12に英語と数学の合計点数を書き込むプログラムを作成しなさい。

Sub makesum()

 For i = 2 To 11

 Cells(i, 6) = Cells(i, 4) + Cells(i, 5)

 あるいは

 Range("F" & i) = Range("D" & i) + Range("E" & i)

 Next i

End Sub

＜例題8＞

　平均点を D12, E12 に書き込みなさい。

Sub calc\_ave()

 sum\_english = 0

 sum\_math = 0

 count = 0

 For i = 2 To 11

 sum\_english = sum\_english + Range("D" & i)

 sum\_math = sum\_math + Range("E" & i)

 count = count + 1

 Next i

 ave\_english = sum\_english / count

 ave\_math = sum\_math / count

 Range("D12") = ave\_english

 Range("E12") = ave\_math

End Sub

　変数countを使わずにave\_english = ave\_english / 10 としても同じ答が得られます。しかし、人数を変更して For i = 2 To 9 のようにしたとき、この解答例のように変数countを導入しておけば、ave\_english / 10 の部分を ave\_english / 8 に書き換える必要がありません。

　今回の例では欠席者は居ませんでしたが、欠席者がおり、欠席者は人数にカウントしたくない場合、次節で習うIf文との組み合わせで、欠席者をノーカウントにできます。

　変数countを導入することにより、プログラムの変更・流用時に、誤りが混入する可能性が低くなります。

　このプログラムには 2, 11, "D12", "E12"という数値がプログラム中に直接書き込まれています。これをマジックナンバーと言い、プログラム中にマジックナンバーを書き込むのは良くないとされています。以下のように変数last\_rowを導入すると、人数が変わったとき、書き換える箇所が1箇所で済みます。

last\_row = 11

　　　　（省略）

For i = 2 to last\_row

　　　　（省略）

Range("D" + CStr(last\_row + 1)) = ave\_english

＜例題9＞

　検算のためExcelのAverage関数を用いて、英語と数学の平均点を求め、D13, E13に入れなさい。

　　　D13に入れる数式　　=AVERAGE(D2:D11)

　　　E13に入れる数式　　=AVERAGE(E2:E11)

# IFの使い方

　条件が成立するか否かによって、処理内容を変えるのが If文です。

＜例題10＞

　合計点が120点以上のとき、合否1の欄に「○」、そうでないとき「×」を書き込みなさい。

学ぶ命令

　If ～ Then ～ Else ～ End If

Sub hantei()

 For i = 2 To 11

 If Range("F" & i) >= 120 Then

 Range("H" & i) = "○"

 Else

 Range("H" & i) = "×"

 End If

 Next i

End Sub

　この例では条件成立、不成立の場合ともに、実行する処理は1行ですが、複数行書くことも出来ます。

　Else 以下が不要の場合は次のように書きます。

 If 条件 Then

 条件が成立したときの処理

 End If

条件には以下のような書き方があります。

　　a = b aとbは等しい

　　a <> b aとbは等しくない

　　a >= b aはbより等しいか大きい

　　a > b aはbより大きい

　　a < b aはbより小さい

　　a <= b aはbより小さいか等しい

（補足）

　a => b あるいは a =< b と書くと、自動的に修正されて、それぞれ a >=b, a <= bとなります。

　2つの条件があり、「どちらかが成立したとき」あるいは「両方が成立したとき」にThen以下の処理を行いたい場合は、それぞれ以下のように書きます。

 If 条件1 Or 条件2 Then

 どちらかが成立したときの処理

 End If

 If 条件1 And 条件2 Then

 両方とも成立したときの処理

 End If

（注意）

　If a = 2 Or 3 Then

という書き方は誤っています。

　If a = 2 Or a = 3 Then

が正しい書き方です。誤った書き方の例では、Excelは2つの条件式「a=2」と「3」の真偽を判定します。「3」はエラーとはならず常に条件成立と見なされるため、結果として条件が常に成立してしまいます。

　プログラムの1行が長くなり、途中で分割したいときは、以下のように、スペースとアンダースコアを書くと、続きを次の行に書くことが出来ます。

If Range("C" & i) >= 60 And Range("D" & i) >= 60 And \_

 Range("B" & i) >= 9 Then

 3つの条件が全て成立したときの処理

End If

＜課題3＞

　合計得点が120点以上を合格とするが、出席日数が8回以下の人は、たとえ120点以上取っていても不合格にし、結果を合否1に書き込みなさい。

＜課題4＞

　出席日数は関係ない。英語、数学ともに60点以上の人が合格とし、結果を合否2に書き込みなさい。

＜課題5＞

　英語、数学について、60点以上とった人の平均点をD14, E14に書き込みなさい。

＜例題11＞

　確認のため、AverageIf関数を使って、60点以上とった人の平均点をD15, E15に書き込みなさい。

　　　D15　　=AverageIf(D2:D11, ">= 60")

　　　E15　　=AverageIf(E2:E11, ">= 60")

# プログラムの作成練習

＜例題12＞

　セルに色を付けなさい。

　「開発」→「コード：マクロの記録」をクリックした後、セルに色を付け、「マクロの終了」をクリックします。

　次にVisual Basic Editorを起動してプロジェクトを見ると（表示されていない場合は、「表示」→「プロジェクト エクスプローラ」）、新しいモジュールが自動生成されており、その中に次のように記録されています。

Sub Macro1()

 Range("B2").Select

 With Selection.Interior

 .Pattern = xlSolid

 .PatternColorIndex = xlAutomatic

 .Color = 13434879

 .TintAndShade = 0

 .PatternTintAndShade = 0

 End With

End Sub

　この例ではセルB2を「左クリック」→「右クリック」→「セルの書式設定」→「塗りつぶし」→「その他の色」で薄い黄色を選択しました。Colorが色を表しています。13434879はRGBのレベルを表しており、B×256×256＋G×256＋Rです。RGBの値はそれぞれ0～255の値をとることができます。13434879 = 204×256×256 + 255×256 + 255なので、R=255, G=255, B=204です。13434879と書かずにRGB(204,255,255)と書くこともできます。

　従って、セルA4を薄い黄色にするには

 Range("A4").Interior.Color = RGB(255, 255, 204)

と書きます。セルA4を塗りつぶしなしにするには

Range("A4").Interior.Pattern = xlNone

あるいは

Range("A4").Interior.Color = xlNone

とします。

＜課題6＞

　出席の列においては9回未満、英語、数学の列においては60点未満、合計得点の列においては120点未満のセルを、薄い黄色（RGB(255,255,150)）で着色するプログラムを作成しなさい。ただし、着色しないセルは「塗りつぶしなし」に設定しなさい。

＜課題7＞

　英語が50～59点の人は補習を実施することにした。補習が必要な人の欄に「○」を記入し、不要な人の欄は空にするプログラムを作りなさい。例えばセルA1を空にするには、Range("A1") = "" と書きます。

＜課題8＞

　数学が40～59点でかつ、出席日数が9回以上の人は補習を実施することにした。補習が必要な人の欄に「○」を記入し、そうでない人の欄は空にするプログラムを作りなさい。

＜課題9＞

　出席点は15回出席したら満点（20点）であり、欠席は2回まで許される。3回以上の欠席は、(欠席回数－2)×3点　を減点する。すなわち、3回欠席は17点、4回欠席は14点・・・となる。ただし、マイナスにはならない。C列に出席点を記入しなさい。

＜課題10＞

　出席点20点、英語40点、数学40点を満点として100点満点で評価した点数をG列に書き込み、60点未満のセルをピンク色（RGB(255,220,255)）で着色しなさい。ただし、60点以上のセルは「塗りつぶしなし」に設定しなさい。

＜例題13＞

　シート「名簿作成」をActiveにして下さい。A列に20人の名前が入っており、薄い黄色（RGB(255,255,200)）に着色されたセルは男性を表しています。

　男性のみを抽出して、C列に並べるプログラムを作成しなさい。

 gyou = 1

 For i = 1 To 20

 iro = Range("A" & i).Interior.Color

 If iro = RGB(255, 255, 200) Then

 Range("C" & gyou) = Range("A" & i)

 gyou = gyou + 1　（イチ　アイではない）

 End If

 Next i

＜課題11＞

　シート「名簿作成2」をActiveにして下さい。A列に20人の名前が入っており、B列はピックアップしたい人は1が入っています。ピックアップしたい人をD列に並べるプログラムを作成しなさい。

＜課題12＞

　シート「名簿作成3」をActiveにして下さい。A列に名前が入っていますが、ところどころ空白のセルがあり、13名の名前があります。この13名をC1～C13に並べるプログラムを作成しなさい。

　セルA1が空セルのとき条件成立なら、以下のように書きます。

 If Range("A1") = "" Then

　セルA1が空セルでないとき条件成立なら、以下のように書きます。

 If Range("A1") <> "" Then

＜例題14＞

　シート「グループ作成」をActiveにして下さい。A列に名前が入っています。4名ずつ取り出して、以下のような名簿を作成するプログラムを作成しなさい。



 gyou = 1

 retu = 3

 For i = 1 To 40

 Cells(gyou, retu) = Range("A" & i)

 retu = retu + 1

 If retu = 7 Then

 retu = 3 あるいは retu = retu - 4

 gyou = gyou + 1

 End If

 Next i

＜課題13＞

　シート「グループ作成2」をActiveにして下さい。B列に1が入っている人だけをピックアップして、C1, D1, E1, C2, D2, E2, C3, D3, E3.... の場所に並べて下さい。

（ヒント）

Ifの中にIfを入れることができます。以下のように書きます。

 If 条件1 Then

 条件1成立の時に実行する文

 If 条件2 Then

 条件2成立の時に実行する文

 End If

 End If

# 簡単なアルゴリズム

　問題を解くための手順をアルゴリズムと言います。

　最小値を求めるアルゴリズムは以下のようになります。

1. 1つめの値を最小値と仮定する。
2. 2つ目以降の値を順番に調べ、現時点での最小値よりも小さい値を見つけたなら、その値を最小値と仮定する。

＜例題15＞

　シート「成績処理」をActiveにして下さい。D16に英語の最小値を書き込みなさい。

 e\_min = Range("D2")

 For i = 3 To 11

 tokuten = Range("D" & i)

 If tokuten < e\_min Then

 e\_min = tokuten

 End If

 Next i

 Range("D16") = e\_min

　最初に「その時点での最小値」をD2と仮定しています。今回の例では問題は生じませんが、この方法は、D2に入っている値が、欠席を表す「欠」のような文字や「-1」のような数値の場合に、うまくいきません。欠席の時は「欠」であるなら、次のようにします。

 find = 0

 For i = 2 To 11

 tokuten = Range("D" & i)

 If tokuten <> "欠" Then

 If find = 0 Then

 e\_min = Range("D" & i)

 find = 1

 Else

 If tokuten < e\_min Then

 e\_min = tokuten

 End If

 End If

 End If

 Next i

 Range("D16") = e\_min

　テストの点数の最大値が100であることが分かっているなら、以下のようなプログラムも可能です（欠席には対応していません）。

 e\_min = 1000 （100以上であれば何でも良い）

 For i = 2 To 11

 tokuten = Range("D" & i)

 If tokuten < e\_min Then

 e\_min = tokuten

 End If

 Next i

 Range("D16") = e\_min

＜課題14＞

　数学の最小値をE16, 最大値をE18, 英語の最大値をD18に書き込むプログラムを作成しなさい。

＜課題15＞

　Excelの関数Min, Maxを使って英語と数学の最小値、最大値を求める数式をD17, D19, E17, D19に記入しなさい。

＜例題16＞

　シート「電気料金」を使います。関西電力の従量電灯A, 従量電灯Bの料金プランは、以下のようになっています（端数切り上げ）。電気料金を計算してB列とC列に記入するプログラムを作成しなさい。そして、横軸を使用電力量、縦軸を料金としてグラフ（散布図）を描き、比較しなさい。

従量電灯A

|  |  |
| --- | --- |
| 最初の15kWhまで | 374円 |
| 15kWh超過120kWhまで | 23円/kWh |
| 120kWh超過300kWhまで | 30円/kWh |
| 300kWh超過分 | 34円/kWh |

従量電灯B

|  |  |
| --- | --- |
| 基本料金 | 389円 |
| 最初の120kWhまで | 21円/kWh |
| 120kWh超過300kWhまで | 25円/kWh |
| 300kWh超過分 | 29円/kWh |

 For i = 2 To 29

' Plan A

 ryokin = 374

 a = Range("A" & i)

 If a > 300 Then

 ryokin = ryokin + (a - 300) \* 34

 a = 300

 End If

 If a > 120 Then

 ryokin = ryokin + (a - 120) \* 30

 a = 120

 End If

 If a > 15 Then

 ryokin = ryokin + (a - 15) \* 23

 End If

 Range("B" & i) = ryokin

' Plan B

 ryokin = 389

 a = Range("A" & i)

 If a > 300 Then

 ryokin = ryokin + (a - 300) \* 29

 a = 300

 End If

 If a > 120 Then

 ryokin = ryokin + (a - 120) \* 25

 a = 120

 End If

 ryokin = ryokin + a \* 21

 Range("C" & i) = ryokin

 Next i