

# Visual Basic によるプログラミング入門

虎 澤 慶 太

## はじめに

情報処理教育センターにおいて情報科学 I の科目でプログラミング入門として、Micro Soft 社製 Visual Basic6.0 (以降 VB6 と省略する) を用いた授業を 2006 年より 8 年にわたって行ってきた。しかし、Windows-XP のサポート終了に伴いこの言語の利用も 2013 年度の授業を最後にせざるを得なくなった。そこで VB6 を使った授業を終了するにあたり、これまで行ってきた情報科学の授業内容をまとめ記録しておくことにした。

## 授業の目的

VB6 は GUI (Graphical User Interface) のフォームに部品を貼り付け、わずかな命令文を記述することで、32Bit 版 Windows 上で実行可能なプログラムを簡単に完成させることができ、プログラミングの初心者にもアプリケーションを作ることができる言語として便利なものである。また Basic 言語 (Beginner's All-purpose Symbolic Instruction Code) という名称自体、「初心者向け汎用記号命令コード」と言う意味である。そのため入門の位置づけの授業にふさわしいと思いこの言語を使った授業をすることにした。

またこの VB6 を使った授業を始めたのは「情報」が高校の必修教科となり、それ以前に行っていたワープロ、表計算、プレゼンテーションソフトを使ったコンピュータ入門の授業が、高校での授業内容とかなりの部分で重複するようになったため、高校で行う授業内容と異なるものをおこなうためでもあった。しかしこの言語は 1998 年に発売された 32Bit 版 Windows 対応

の最終バージョンであり、2008年4月にはサポートも終了している。この授業を始めた2006年にはVB6のソフトウェアはすでに販売が終了しており、授業の準備は情報処理教育センターのクライアント室に行っておこなう必要があった。

## 授業の内容

授業の概略は次のようである。

まず、始めにVB6の起動方法と終了時に注意すべき点をプリントにまとめ配布した。そして、続く2回の授業でVB6を使うことで簡単にアプリケーションを作ることができることを示し、なおかつ初回の授業で注意した終了時に行う一連の操作について復習を行った。次いで、授業の中心となるデータの処理プログラムを作るために、データの入出力とファイルへの保存と読み出しのためいわゆる「memoプログラム」というアプリケーションを作成した。そして、そのアプリケーションにコマンドボタンの追加や、それに伴う新しいサブルーチンを追加、拡張することで徐々にアプリケーション中で行う処理を増やしていった。

続いて、繰り返し命令のDo LoopとFor Nextの使い分けと処理の違いについて学び、再帰的プログラムを学んだ。最後に定期試験の代わりとして簡単な文字の処理を行うことで暗号化した問題文を解説して課題とした。

では、授業内容を授業をおこなった順に列挙していく。なお、これらの授業内容は少しずつ改良を加えていった結果として2013年春学期に行ったものである。何回目の授業であるか判るように、番号を始めに付記している。

### 1. 時刻と日付の表示プログラム

VB6の起動方法と終了時に注意すべき点をプリントにまとめ、簡単なアプリケーションの作成をした。GUIのフォームにコマンドボタンと表示部であるラベルの部品を貼付け、フォームのデザインを行う。そしてコマンドボタンが押下されたときに行うサブルーチンを「コード」として入力した。さらにプロパティの設定とその意味について学んだ。この回のプログラムは、「確認」ボタンを押下すると二カ所設定した日付と時刻の表示部に、ボタンを押した時点の日時と時刻を表示するものである。

## 2. デジタル時計の作成

1 回目で作ったプログラムをもとに、デジタル時計を作成した。これは受講生ごとに写真等の画像を用意させ、各自オリジナルのデジタル時計を作成した。この回のプログラムでは、前回おこなった「確認」ボタンを押す代わりに「タイマー」機能を利用してあらかじめプロパティの設定で指定した時間ごとにサブルーチンを起動して表示を更新させるようにしたものである。

## 3. ストップウォッチ機能付きデジタル時計の作成

2 回目の授業で作成したデジタル時計にストップウォッチ機能を追加した (図 1)。

これには本来表示部として使うラベルの部品を、プロパティ機能により「透明化」してデジタル時計の画像の上に配置することで画像の特定の位置が見えないボタンとして働くように工夫したものである。例としてあげたものでは、犬の鼻の周辺がスタートボタン、舌の周辺がストップボタン、そしてよだれかけ周辺が表示のクリアボタンとなっている。また、ボタンとなっていることが解りやすいように、マウスカーソルの形状が鼻の周辺では砂時計に、舌の周辺では禁止マーク、そしてよだれかけ周辺ではクロスカーソルへと変化するようにしてある。



図 1

(スクリーンショットでは、マウスカーソルが表示されないため写っていない)

この授業は、1 回目、2 回目の授業のまとめであると同時に簡単な作業でちょっとしたプログラムを作ることができる事を示すとともに、後にプログラムが計算を行うのに必要とした時間を測定するためのルーチンを伏線として与えておく事が目的である。

## 4. memo プログラムの作成

データ処理のためのプログラムを行う前段階として、データの入出力と保存・作成のための基本的な処理を行う事のできる memo プログラムを作成した (図 2)。この回からはフォームのデザイン、基本的なサブルーチンのコードや学生が間違えやすいコードなどは「ひな形」として授業用サーバーに置いたものを、学生が各自のマイドキュメントにコピーしてから操作す

るように指導した。(図3) また、この授業では動作に必要なコードをプリントにして配布した。

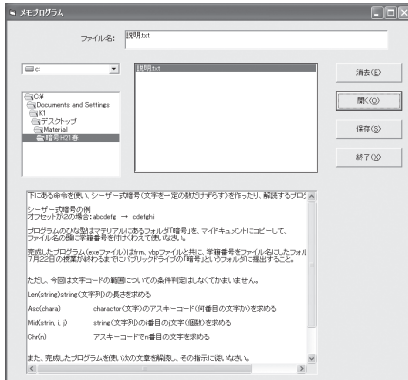


図 2

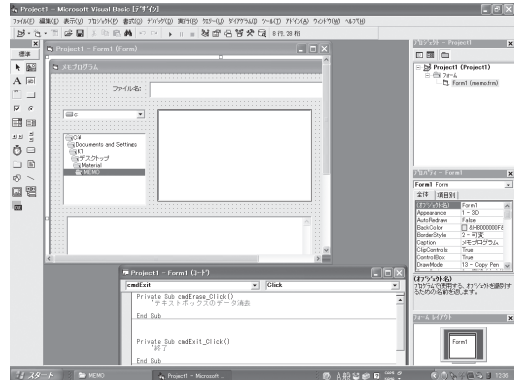


図 3

このプリントでは「ひな形」のコード部分から虫喰いにした部分を太字で表示してあり、また虫喰い部分の前にはどのような作業をするためのコードであるかを示すコメント文を付けて入力を助けている。以下にコードの一部を示す。

Private Sub cmdErase\_Click()

‘テキストボックスのデータ消去

**txtFilename.Text = ""**

**txtDisplayBox.Text = ""**

End Sub

Private Sub cmdEnd\_Click()

‘終了

**End**

End Sub

以下略

## 5. 判定プログラムの作成

300人分の架空の学籍番号と、それぞれに点数を乱数で作成したものを CVS 形式のテスト

データとして用意し、各自の点数を AA から E までのアルファベットによる成績判定に変換して表示するプログラムを作成した（図 4、図 5）。点数により複数の判定値から選択するため、Select Case 文を使った分岐を学んだ。

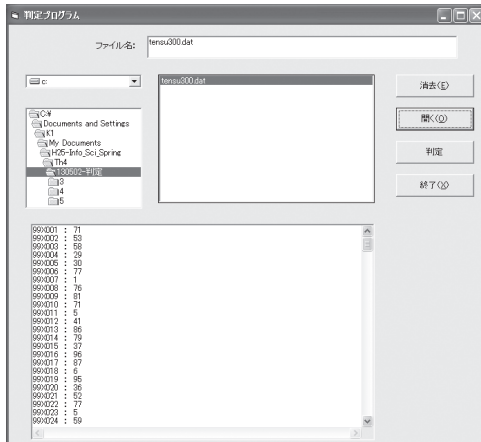


図 4

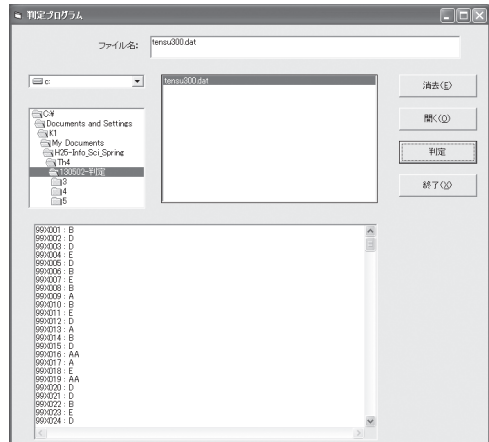


図 5

以下にコードの一部を示す。

```

'判定をする
Select Case Val(temp(i,2))
    Case 90 To 100
        temp(i,3) = "AA"
    Case 80 To 89
        temp(i,3) = "A"
        (中略)
End Select
以下略
    
```

この回のひな形から「保存」のコマンドボタンをのぞいたものを使った。これは、この授業内容を始めた頃、「判定」した後のデータを保存してしまい、それ以降テストデータを使う事ができなくなってしまう学生が後を絶たず、基となるデータの改変を防ぐためである。

## 6、7. 並べ替えプログラムの作成（バブルアップ法）

5 回目の授業で使った架空の成績を、点数の順で並べ替えるプログラムを作成した（図 6、図 7）。この授業では、降順に並べ替えるルーチンのみを与えた。

以下にコードの一部を示す。

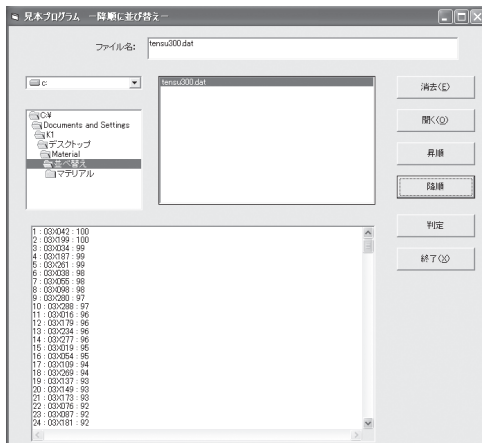


図 6

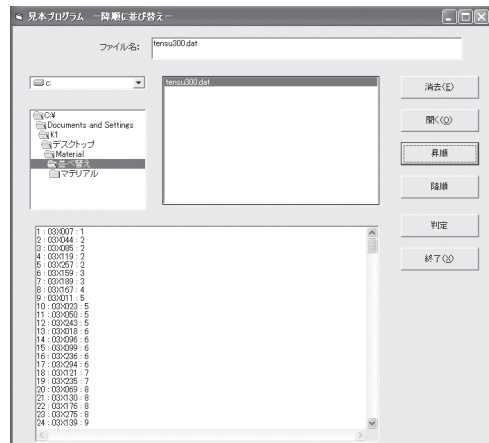


図 7

Do

F = 0

For i = i To c - 1

‘点数を較べる（どっちが大きいかな？）

If Val(a(i + 1, 3)) > Val(a(i, 3)) Then

‘後の点数が大きければデータを一時退避する

g(1, 2) = a(i + 1, 2)

g(1, 3) = a(i + 1, 3)

(中略)

F = 0

End If

Next i

‘入れ替えがなくなるまで繰り返す

Loop Until F = 0

以下略

ここでは、それぞれの命令がどのような作業をしているかコメントとして明記した。この授業では「昇順」に並べ替えるためのコマンドボタンの追加と昇順に並べ替えるためのルーチンも作成する事に加え、2回の授業を充てた。この並べ替えプログラムでは隣り合うデータ同士を比べ、入れ替えがなくなるまで繰り返す、俗にバブルアップ法という効率の悪い方法を使用した。しかし、この考え方は非常に簡単なものなので並べ替えの原理を理解するためにはふさわしいと考え、これを使用した。

## 8. 九九プログラムの作成 (For 文による繰り返し)

九九の表を「横何番目まで」「縦何段目まで」を任意に入力し、表を出力するプログラムを作成した (図8)。

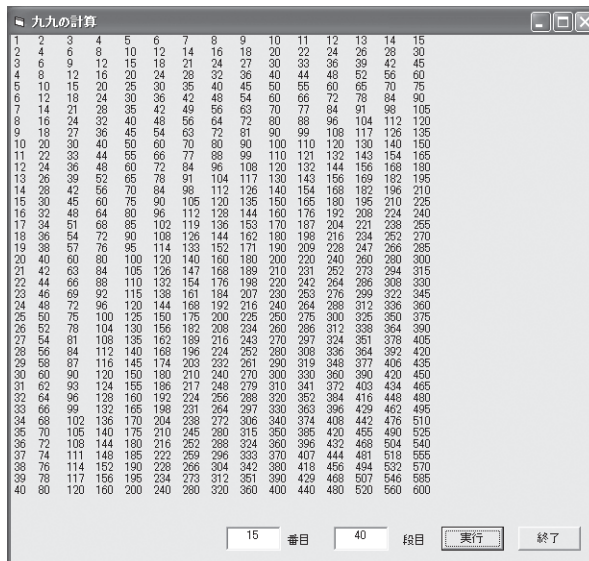


図 8

For 文を使い、繰り返しを入れ籠にする仕方を学んだ。この時の表示方法は、それまでに使ってきた TextBox とは違い、そのまま Form 内に表示する方法を使った。この回のひな形は「何段目」と縦の範囲を問い、横 9 列までを計算し出力するものを与えた。学生は横「何番目」までかをテキストボックス txtYoko に入力し、その数値を使用するために繰り返しの範囲について書き換えるようにした。また、コメント中にその行でおこなう命令を記述し、下線部が入力すべき所である事を示した。以下に変更前のひな形のコードの一部を示す。この回からコメントはその行に付加した。

```
Private Sub cmdJikkou_Click()
    For i = 1 To txtTate
        For j = 1 To 9
            Form1.Print ____ ; Tab(6 * j);           ‘i * j を 6 桁ごとに表示する
        Next j
        Form1.print                                     ‘改行する
    Next i
End Sub
```

今回使用した表示方法では、TextBox の場合と違い print 命令の後に何も付加しないと自動的に改行してしまう。そのため自動改行を防ぐ命令（セミコロンを print 命令に付加する）を付け加えながらコードを書いていく必要がある。また、このプログラムでは桁をそろえた表示にするため、参考文献 1 にある書式を参考にした。

## 9. 九九プログラムの作成 (Do-Loop 文による繰り返し)

8 回目と同じ九九の表を出力するプログラムだが、繰り返しの仕方を Do Loop 文で行った。この繰り返しでは For 文と異なり、While/Until を使い条件判断を繰り返しの始めか終わりで行う事により、より細かに制御ができる点で有用である。今回は繰り返しの終わりに判断をする事にした。以下にひな形のコードの一部を示す。

```
Private Sub cmdJikkou_Click()
    Dim __ As _____ ‘i を整数とする
    Dim __ As _____ ‘j を整数とする
```



```

i = __                ' i を 1 にする
Do
    j = __            ' j を 1 にする
    Do
        Form1.Print ____; Tab( 6 * j);    ' i * j を 6 桁ごとに表示する
        j = ____        ' j を 1 増やす
    Loop Until _____    ' j が txtYoko より小さい間
    Form1.print        ' 改行する
    i = ____            ' i を 1 増やす
Loop Until _____    ' i が txtTate より小さい間
Form1.print            ' 改行する
End Sub
    以下略

```

## 10. フィボナッチ数列プログラムの作成

次の回に行う「再帰的プログラミング」のために、まず For 文を使った通常の繰り返しによるフィボナッチ数列の計算、表示をするプログラムを作成した。ちなみに、フィボナッチ数列とは「1、1、2、3、5、8、13、21、34、55、、、、」というもので、最初の2つを1とし、以後、前の2つを足し合わせたものを次の数としていくものである。この回のひな形は歯抜けのコードと、プリントで1番目から10番目のフィボナッチ数を表示するものを与えた。完成形はプリントを参考にして45番目までの任意のフィボナッチ数を表示させるものにコードを書き換えさせるものとした。

## 11. フィボナッチ数列プログラムの作成（再帰的プログラミング）

前回行った通常の繰り返しによるフィボナッチ数列の計算を、計算の定義から再帰的に求める方法を学んだ。この再帰的プログラミング法は計算の定義をすれば、引数を与えて計算を呼び出すだけで答えを求められ、直感的にプログラムができる方法である。ただ、再帰的計算には1つ次の答えを求めるためにはそれまでの計算に必要とした時間の、およそ倍の時間が必要となる。また、必要とするメモリも等比級数的に必要量が増えていく。そのため、実際のプログラミングにおいては再帰的プログラミングは避けるべきものである。このことを学生に実体

験させるため、「通常の計算」(図9)と「再帰的計算」(図10)の2種類のコマンドボタンをフォームに作り、3回目の授業で使った時間計測のための命令を組み込む事でそれぞれの計算にかかった時間を計測し表示した。

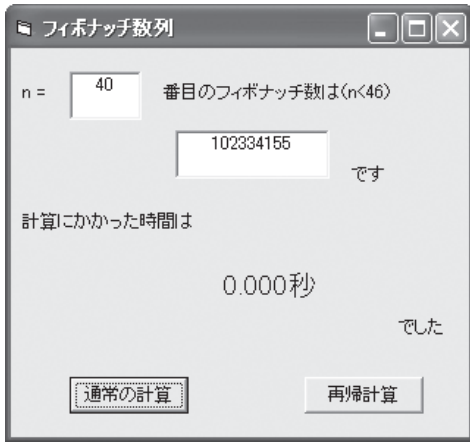


図9

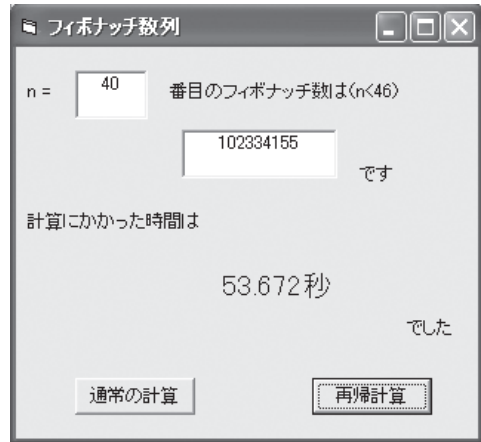


図10

図11は再帰的計算をおこなう部分のコードを示した。

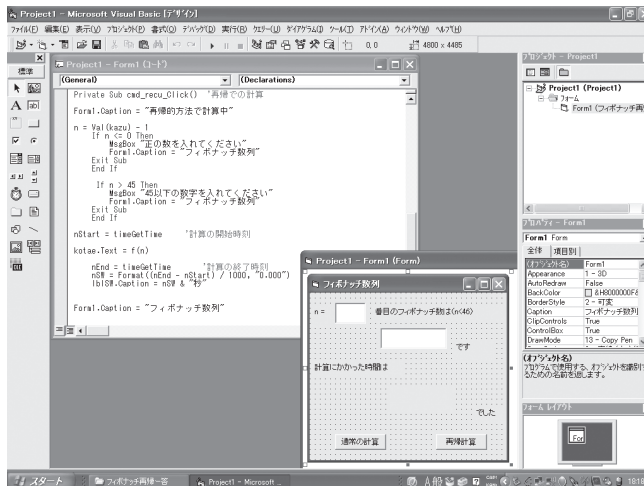


図11

## 12. 並べ替えプログラムの作成（コムソート法）

6、7回目に行ったバブルアップ法による並べ替えをもとに、データの比較対象を隣接するデータから、全データ数の1.3での除数を変数名 `gap`（但し整数部のみ）として、`gap` だけ離れたデータと比較する。次の繰り返しでは `gap` の1.3での除数を新たな `gap` として次の比較を行っていく。この方法をコムソート（comb sort）法と言い、バブルアップ法と比べ50倍以上高速に並べ替えることができる。それまでは300人分のデータであったがコムソート法での並べ替えが100分の1秒以上になるようにするため、データを1000人分に増やして使った。11回目と同様にしてバブルアップ法（図12）とコムソート法（図13）での実行時間を計測するようにした。

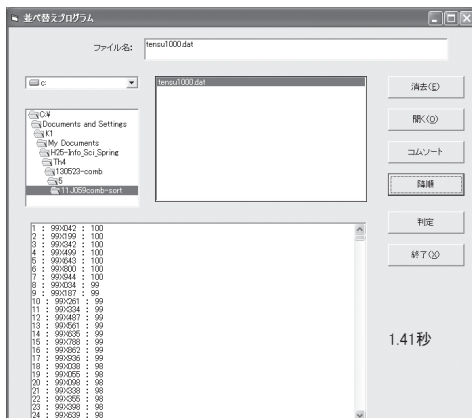


図12

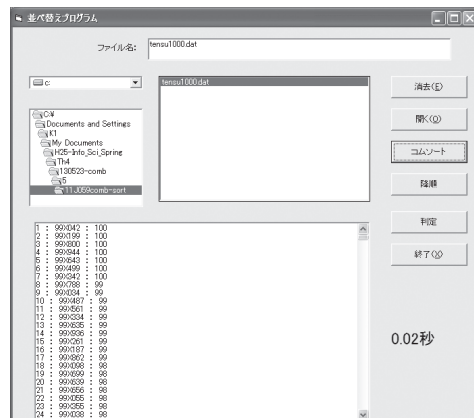


図13

## 13. ハノイの塔プログラムの作成（再帰的プログラミング2）

11回目に行った再帰的プログラミングであるが、関数の定義にはもう1つの定義の仕方がある事を示した。また、すでに再帰的プログラムは実用に向かないものである事を学んだ訳であるが、ゲームの一種である「ハノイの塔」を計算により解く場合には、再帰的プログラミングの方法を使う事により極めて簡単にプログラムを記述し解く事が可能となる。これは再帰的プログラミングが有効である、数少ない例のひとつである。ひな形は歯抜けにしてあり、学生にはプリントとして Tiny Basic で書かれたコードを示し、二つのコードを比較し歯抜けの部分を補うように指導した。以下に5枚のディスクでおこなう場合の Tiny Basic のコード（参考文献2）を示す。

‘実行部分

Call Hanoi( 5, “A” , “B” , “C” )

End

‘手順の定義

Sub Hanoi( n, A\$, B\$, C\$ )

    If n > 0 then

        Call Hanoi( n - 1, A\$, C\$, B\$ )

        Print n; “番の円盤を” ; A\$; ” から” ; B\$; “へ移動”

        Call Hanoi( n - 1, C\$, B\$, A\$ )

    End If

End Sub

図にはディスクが9枚（図14）と10枚（図15）の場合の実行例が示してある。このようにディスクが1枚増やすごとにおよそ4倍の計算時間が必要になる。

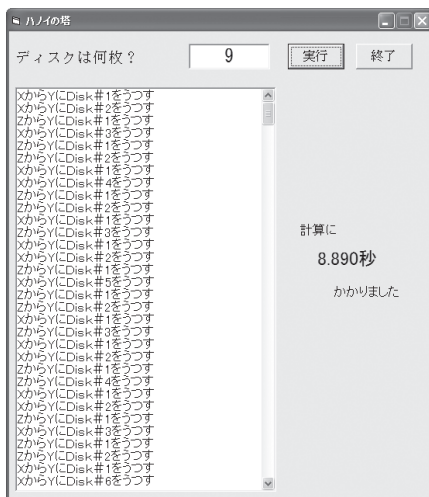


図 14

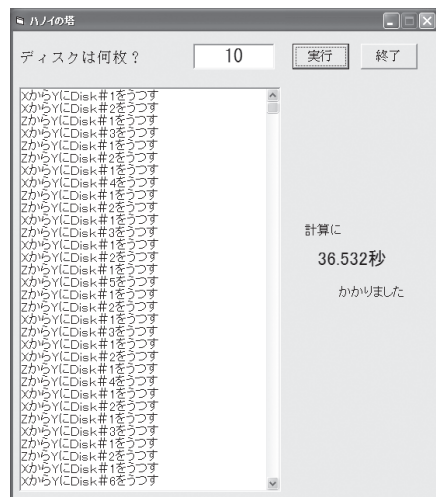


図 15

## 14.15. シーザー式暗号プログラムの作成

この授業は定期試験の代わりとなる内容として行った。古典的な暗号化法の1つであるシー

ザー式暗号について簡単な説明とプログラム中に使うべき VB の命令を示し、かつ参考となる WEB サイトの探し方を明記したファイルを学生に示した。さらにこのファイルの最後には最終課題である暗号化した文章が記載されており、完成したプログラムを使いこの暗号文を解読してその内容の問題（SPI の練習問題）を解き、虎澤に口頭で答えを伝える事で試験の代わりとした。

実際のファイルは以下のようなものである。

---

ファイルここから

---

下にある命令を使い、シーザー式暗号（文字を一定の数だけずらす）を作ったり、解読するプログラムを作りなさい。

シーザー式暗号の例

ずらす量が 2 の場合：abcdefg → cdefghi

プログラムのひな型はマテリアルにあるフォルダ「暗号」を、マイドキュメントにコピーして、ファイル名の頭に学籍番号を付けかわえて使いなさい。  
完成したプログラム（exe ファイル）は frm、vbp ファイルと共に、フォルダ名「学籍番号＋暗号」に入れ、春学期の授業が終わるまでにパブリックドライブの「暗号」というフォルダに提出すること。

今回は文字コードの範囲についての条件判定はしなくてかまいません。

プログラム中で使うと良い命令は次の通りです。

Len(string) string（文字列）の長さを求める

Asc(chara) charactor（文字）のアスキーコード（何番目の文字か）を求める

Mid(string, i, j) string（文字列）の i 番目の j 文字（個数）を求める

Chr(n) アスキーコードで n 番目の文字を求める

また、完成したプログラムを使い次の文章を解読し、その指示に従いなさい。  
オフセットは - 9 ～ 9 のあいだです。

~~~~~ ここから ~~~~~  
いれ灰巖と 211 仁ん耐賞ぬ課沙んじだごわ。ザービイヴんじ  
だごとはいれ仁ぼ 39 仁。ポトイーポーナーんじだごどはいれ仁  
ぼ 56 仁。チュピヴケはむんじだごどはいれ仁ぼ 52 仁とじだ、  
ぞはうせろやじだごどはにう仁きうにうどぎ。ザーブイヴどポトイ  
ーポーナは凌朋んじだごどはいれ仁ぼ伽仁とずが、  
~~~~~ ここまで ~~~~~

---

ファイルここまで

---

また、完成したプログラムを使い上にある暗号文を解読すると図16のようになる。

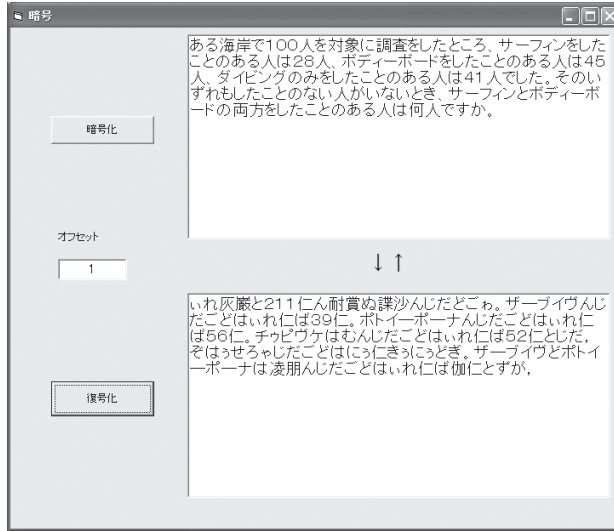


図16

## まとめ

これらの課題を通してプログラムがどのようなものであるかを実感し、かつプログラミングに必要な論理的思考を身につける事が、少ないながらもできたのではないかと思う。受講生が文系の学生である事から「不等号」という言葉を始めとした数学用語は仕方がないとしても、エクスクラメーション・マーク、ダブルクォーテーションや、コロンといった高等学校以前に習うであろう英記号を知らない学生が多く見受けられるようになり、それぞれを「ビックリマーク」「ちょんちょん」「点々」等と言い替えなければならない。また、8回目の授業で使用した歯抜けのコードで、

```
Form1.Print ____; Tab(6*j); 'i*jを6桁ごとに表示する
```

とあるのに、 $i*j$  を下線部に入力する事が理解できない学生が少なからず見られた。これは9回目以降の授業でも同じ学生でこの傾向が見られた。これは、そこに書かれている文章の意味を読み取り、次の行動につなげるという作業を十分に訓練されていないことを示している。これらのことから、いわゆる知育をふくむ基礎的な学習能力の涵養が必要であると思われる。

### 参考文献

1. 戸川隼人「ザ・Visual Basic」サイエンス社 (1999)
2. 古川直樹・五百蔵重典「コンピュータ言語」日経BPソフトプレス (2004)