

【A-4】IchigoJam でトランプゲームを作る（4K 版）

●今回の目標

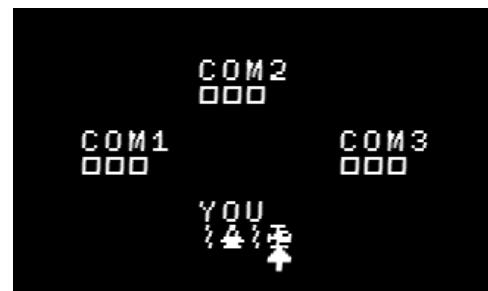
IchigoJam で動くトランプゲームを作ります。

ただし普通のトランプだと、カードの枚数が多かったり、スーツ(マーク、♠♥♦♣)があつたりして、処理が面倒です。

今回はカードの種類が違う、シンプルな「イチゴトランプ」ゲームを作ります。

ルールは以下のとおりです。

- カードは、11 種類の絵柄カード×4 枚、イチゴカード×3 枚、合計 47 枚です。
- プレイヤーは、自分(YOU)と COM1～3 の 4 人です。
- 各人に手札が 3 枚ずつ配られます。残りのカードは山札となります。
- 自分のターンからスタートして、時計回りの順にプレイします。
各人のターンでは、山札から 1 枚カードを引いて手札を 4 枚にし、どれか 1 枚を山札に捨てて再び手札を 3 枚にします。
- 山札から 1 枚カードを引いた時に、手札の中で同じ絵柄が 4 枚そろえば、そのプレイヤーが勝ちとなり、ゲームが終了します。
- 同じ絵柄が 2 枚・3 枚とそろっても、イチゴカードを引いてしまうと、自分の手札全てが山札と自動的に入れ替わり、バラバラになってしまいます。



●プレイ画面を表示する

まず、プログラムの初期設定をした後、画面にプレイヤーの名前を表示します。
あとのことを考えて、表示部分はサブルーチンにします。

10 ' *Ichigo Trump	コメントで、プログラムのタイトルを入れる
20 CLV:CLS	変数をクリアする:画面をクリアする
30 LET [90],5,6,0,3,5,0,10,3	配列変数[90]から、プレイヤー表示の座標を設定
100 FOR P=0 TO 3	プレイヤー変数 P を 0～3 でくりかえし
150 GSB 300	表示サブルーチンと呼ぶ
160 NEXT	くりかえしここまで
290 END	プログラム終了
300 ' @PRTCARD	ここから画面表示サブルーチン
310 Z=90+P*2:X=[Z]:Y=[Z+1]	配列変数から座標 X,Y を設定
320 LC X,Y:IF P=0?"YOU" ELSE?"COM";P	
390 RTN	メインプログラムへもどる
	P=0 の時は自分、その他は COM1～3 を表示

入力できたら、「RUN」で実行してみましょう。
画面がクリアされて、プレイヤーの名前「YOU」「COM1」「COM2」「COM3」が表示されます。

※プログラムが終了する時、「COM1」の下に「OK」が表示されます。今回は気にしないことにします。



30 行では、LET (レット) 命令を使って、配列変数に画面表示の座標を設定しています。

```
30 LET [90],5,6,0,3,5,0,10,3
```

配列変数[90]から順番に、自分(YOU)の X,Y 座標、COM1 の X,Y 座標…を設定します。

90	91	92	93
YOU:X座標	YOU:Y座標	COM1:X座標	COM1:Y座標
94	95	96	97
COM2:X座標	COM2:Y座標	COM3:X座標	COM3:Y座標

300 行からのカード表示サブルーチンでは、変数 P を 0～3 で変化させて、FOR～NEXT 命令でくり返して表示しています。

```
300 / @PRTCARD
310 Z=90+P*2:X=[Z]:Y=[Z+1]
320 LC X,Y:IF P=0?"YOU" ELSE?"COM";P
360 RTN
```

310 行では、「**Z=90+P*2**」として、変数 P から配列の番号 Z を計算しています。

- P=0 の時 → $Z=90+0*2=90$ → $X=[90]:Y=[91]$
- P=1 の時 → $Z=90+1*2=92$ → $X=[92]:Y=[93]$
- P=2 の時 → $Z=90+2*2=94$ → $X=[94]:Y=[95]$
- P=3 の時 → $Z=90+3*2=96$ → $X=[96]:Y=[97]$

これで、30 行で設定した X,Y 座標を取り出しています。

320 行では、「**LC X,Y**」でカーソルを移動させた後、IF 命令の条件判断で、P=0 の時は「YOU」(自分)、P=1～3 の時は「COM1」～「COM3」を表示しています。

「**LC**」は「**LOCATE**」、「**?**」は「**PRINT**」の略です。

30 行で、LET 命令で配列変数に設定している座標を変えると、各プレイヤーの表示位置が変わります。いろいろ変えて試してみましょう。

●カードを配る

最初にカードをシャッフル(混ぜる)して、各プレイヤーに配ります。
この「カードをどう管理するか」が、トランプゲームのプログラムのポイントです。

基本的には、配列変数を使ってカードを管理します。
イチゴトランプではカードが全部で 47 枚あるので、配列変数[0]～[46]を使います。
11 種類の絵柄は、文字コード 235～245 の 11 個を使います。

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
230							♠	♣	♥	♠
240	♠	♣	♥	♠	♣	♥	♠	♣	♥	♠

まず、配列変数[0]～[43]に、順番に絵柄カード 44 枚をセットしましょう。
(3 枚のイチゴカードは、あとでセットします。)
そして、各プレイヤーの手札を、配列変数[50]～[69]にセットします。

- [50]～[54] 自分(YOU)
- [55]～[59] COM1
- [60]～[64] COM2
- [65]～[69] COM3

```

10 ' *Ichigo Trump
20 CLV:CLS
30 LET [90],5,6,0,3,5,0,10,3
40 FOR N=0 TO 3      カード枚数 N を 1～4 でくりかえし
50 FOR S=235 TO 245  カード種類 S を 235～245 でくりかえし
60 [M]=S:M=M+1      カード配列にカードを設定
70 NEXT:NEXT        くり返しここまで
100 FOR P=0 TO 3
110 Q=50+P*5         各プレイヤーのカード配列番号 Q を計算
120 FOR N=0 TO 2     カード 1 枚目～3 枚目をくりかえし
130 M=M-1:[Q+N]=[M]:[M]=0  カードを 1 枚取り出してプレイヤーに配布
140 NEXT
150 GSB 300
160 NEXT
290 END

```

カード表示サブルーチンも書きかえて、カードがどう配布されたか表示してみましょう。

```

300 ' @PRTCARD
310 Z=90+P*2 : X=[Z] : Y=[Z+1]
320 LC X,Y : IF P=0 ? "YOU" ELSE ? "COM" ; P
330 LC X,Y+1 プレイヤー名の次の行にカーソルを移動
340 FOR N=0 TO 3 カードを1枚目～4枚目までくりかえし
350 F=[Q+N]
360 ?CHR$(F) ; カード配列から1枚読み出して表示
370 NEXT : RTN くりかえしここまで

```

プログラムを実行してみましょう。

4人のプレイヤーの手札が表示されます。
ただ順番にカードを配っているだけなので、プログラムを何回実行しても同じ手札になります。

※他のプレイヤーの手札が見えてしまうとゲームになりませんが、動作確認のために今は表示しています。あとで他のプレイヤーの手札は隠します。



プログラムの内容を説明します。

まず40～70行では、配列変数に絵柄カードを配置しています。

```

40 FOR N=0 TO 3
50 FOR S=235 TO 245
60 [M]=S : M=M+1
70 NEXT : NEXT

```

40行のFOR命令は、同じ絵柄のカードが4枚ずつあるので、それをくり返し、
50行のFOR命令は、カードの絵柄を文字コード235～245でくり返しています。
60行で、変数Mで配列の番号を指定しています。20行のCLVで変数をクリアしているので、Mは最初は0です。「[M]=S」で配列[M]にカードを設定した後、「M=M+1」でMを1増やしているため、Mは0,1,2,3…と変化していきます。

この結果、配列変数の値は以下のようになります。

配列	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	...
値	235	236	237	238	239	240	241	242	243	244	245	235	...
絵柄													...

100～160 行では、各プレイヤーにカードを配って、それを画面に表示しています。

```

100 FOR P=0 TO 3
110 Q=50+P*5
120 FOR N=0 TO 2
130 M=M-1:[Q+N]=[M]:[M]=0
140 NEXT
150 GSB 300
160 NEXT

```

100 行の FOR 命令は、プレイヤー変数 P を 0 (YOU) ～3 (COM3) までくり返しています。
 110 行「Q=50+P*5」で、各プレイヤー用の配列の先頭番号 (50,55,60,65) を計算します。
 プレイヤーのカード配列の番号は、以下のようになります。

50	51	52	53	54
YOU:カード0	YOU:カード1	YOU:カード2	YOU:カード3	YOU:カード4

55	56	57	58	59
COM1:カード0	COM1:カード1	COM1:カード2	COM1:カード3	COM1:カード4

60	61	62	63	64
COM2:カード0	COM2:カード1	COM2:カード2	COM2:カード3	COM2:カード4

65	66	67	68	69
COM3:カード0	COM3:カード1	COM3:カード2	COM3:カード3	COM3:カード4

なお、手札は最大でも 4 枚なので、カード配列は 1 人あたり 4 個で済みそうですが、あとでカードを処理する都合で 5 個ずつ割り当てています。

120 行の FOR 命令は、カードを 3 枚ずつ配るので、N を 0～2 までくり返しています。

130 行では、変数 M を 1 減らして、プレイヤーの配列へカード配列の値を移して、カード配列の方は 0 にしてクリアしています。(＝カードの山の一番上から 1 枚ずつ配る動作)

140 行で、N のループを終わらせた後、150 行でカード表示サブルーチンを呼んで、配ったカードを表示しています。

カード表示サブルーチンの追加部分では、プレイヤーのカード配列を元にカードを表示しています。

```
330 LC X,Y+1
340 FOR N=0 TO 3
350 F=[Q+N]
360 ?CHR$(F);
370 NEXT:RTN
```

カード配列の値は文字コードで入っているので、それを変数 F に読み出して、「?CHR\$(F);」でカードを表示しています。

なお、FOR 命令のくり返しでカードを 4 枚表示していますが、カードを 3 枚しか持っていない時は 4 枚目のカード配列の値が「0」になっているので、表示も「?CHR\$(0);」となり、空白が表示されます。

●カードをシャッフルする

このままだと、各プレイヤーの手札や山札が毎回同じパターンになってしまうので、ゲームになりません。

配る前に、カードをシャッフル(混ぜる)することになります。

シャッフルする方法はいろいろありますが、今回は「ランダムに2枚選んで、そのカードを交換する」「それを何度もくり返す」ことにします。

また、後のプログラムの都合で、カードをシャッフルする処理をサブルーチンにします。

```
40 FOR N=0 TO 3
50 FOR S=235 TO 245
60 [M]=S:M=M+1
70 NEXT: NEXT
```

```
80 GSB 400
```

シャッフルサブルーチンを呼ぶ

```
100 FOR P=0 TO 3
... (中略) ...
390 RTN
```

```
400 / 0SHUFFLE
```

```
410 FOR N=1 TO 100
```

シャッフルを100回くり返し

```
420 A=RND(M):B=RND(M)
```

乱数でカードを2枚選ぶ

```
430 D=[A]:[A]=[B]:[B]=D
```

2枚のカードを入れ替える

```
440 NEXT: RTN
```

ここまでくり返し:もどる

プログラムを実行してみましょう。

今度はカードがランダムになり、実行するたびにプレイヤーの手札が変わります。



400～440 行のカードをシャッフルするサブルーチンでは、「2枚のカードを選んで交換」という動作を100回くり返しています。

410 行の FOR 命令のくり返しを多くすれば、それだけカードがランダムになりますが、処理時間がかかります。

くり返しの回数を変えて、手札がどうなるか試してみましょう。

●イチゴカードを追加する

通常の絵柄カードを配れたので、これに 3 枚のイチゴカードを追加します。

4 人のプレイヤーにカードを配った後、残った山札にイチゴカードを追加して、再度シャッフルすることになります。

(最初からイチゴカードを入れてプレイヤーに配ってしまうと、いきなり誰かがイチゴカードを引く場合があり、プログラムが面倒になるからです)

```

100 FOR P=0 TO 3
110 Q=50+P*5
120 FOR N=0 TO 2
130 M=M-1:[Q+N]=[M]:[M]=0
140 NEXT
150 GSB 300
160 NEXT
170 FOR N=1 TO 3
180 [M]=255:M=M+1
190 NEXT:GSB 400
290 END

```

イチゴカード 3 枚分くり返し

カード配列の最後にイチゴカードを追加、M を 1 増やす

くり返しここまで:シャッフルサブルーチンを呼ぶ

イチゴカードは、文字コード 255 のイチゴのキャラクター「」を使います。

プログラムを実行しても、各プレイヤーの手札はこれまでと変わりません。

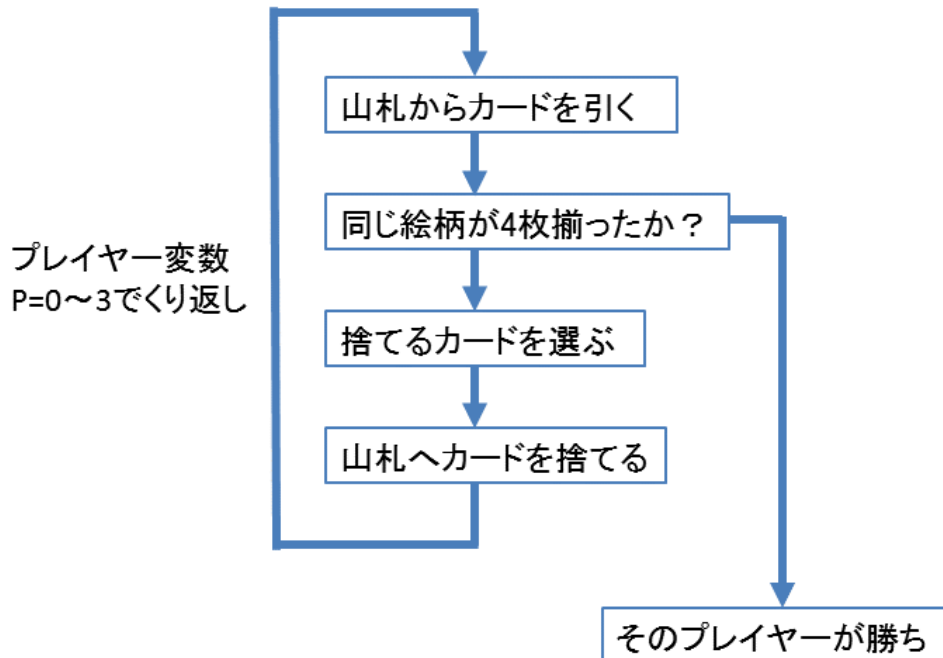
(イチゴカードは山札の中にあります)

この後、ゲームをプレイしていくと、イチゴカードを引いてくるはずです。

「SAVE 0」でプログラムを保存しておきましょう。

●カードを引く

カードの設定ができたので、いよいよゲームプレイ部分を作っていきます。
ゲームプレイの流れ(フローチャート)を考えてみます。



まず自分(YOU)のターンで、山札からカードを1枚引きます。

プレイヤー変数 P を 0 にする

```

190 NEXT:GSB 400:P=0
200 / @GAMELOOP
210 Q=50+P*5:D=RND(M):[Q+3]=[D]:[D]=0
220 GSB 300
280 WAIT 60:P=(P+1)%4:GOTO 200
290 END
  
```

山札からランダムに1枚カードを引き、
プレイヤーカード配列の4枚目に入れる。
引かれた山札は 0 でクリアする

引いた状態を画面に表示する

プレイヤー変数を更新:ループ先頭へもどる

プログラムを実行してみましょう。

自分(YOU)がカードを1枚引きます。

順番に他のプレイヤーもカードを1枚引きます。

「SAVE 0」でプログラムを保存しておきま
よう。



●捨てるカードを選ぶ

カードを1枚引いて4枚にした所で、1枚捨てるカードを選びます。

(※同じ絵柄カードが4枚揃ったかどうかのチェックは、後回しにします)

自分(YOU)の操作と、COM1～3のカード選択はやることが違うので、それぞれサブルーチンと呼ぶ形にします。

まず、ゲームループのプログラムを書きかえます。

```

200 ' @GAMELOOP
210 Q=50+P*5:D=RND(M):[Q+3]=[D]:[D]=0
220 GSB 300
240 Z=90+P*2:X=[Z]:Y=[Z+1]:C=3
250 IF P=0 GSB 1100 ELSE GSB 1200
260 [D]=[Q+C]:FOR N=C TO 3:[Q+N]=[Q+N+1]
280 WAIT 60:P=(P+1)%4:GOTO 200
290 END

```

表示座標の呼び出し:デフォルトの捨てるカードを4枚目にする

カード選択

山札へカードを捨てる:捨てたカード分を詰める:カードを表示

カード選択のサブルーチンを作ります。

```

1100 ' @YOUPLAY
1110 LC X+C,Y+2:?" ";
1120 K=INKEY():C=C-(K=28)*(C>0)+(K=29)*(C<3)
1130 LC X+C,Y+2:?"CHR$(226);
1140 IF K!=32 GOTO 1100
1150 BEEP:LC X+C,Y+2:?" ";:RTN
1200 ' @COMPLAY
1210 C=RND(4):RTN

```

自分(YOU)の思考サブルーチン

カード選択カーソルを消す

左右カーソルキーで、カード選択カーソルの座標を変更

カード選択カーソル「↑」を表示

スペースキーが押されていなければもどる

COMの思考サブルーチン

BEEP音:選択カーソルを消す:もどる

とりあえず乱数でカードを1枚選ぶ

「RUN」で実行します。

プレイヤーは左右カーソルキーでカードを選んで、スペースキーで捨てられます。

他のCOMプレイヤーは、ランダムにカードを選んで捨てます。

いちおうトランプゲームの動きになります。



自分が捨てるカードを選択するプログラム(スロット1)では、「↑」のカーソル位置(変数 C)をキー入力で変化させています。

```
1110 LC X+C,Y+2:?" ";
1120 K=INKEY():C=C-(K=28)*(C>0)+(K=29)
*(C<3)
1130 LC X+C,Y+2:?"CHR$(226);
```

1120 行では INKEY(インキー)関数を使って、キー入力を読み取っています。

INKEY()

返り値	押されているキーの文字コードが返ってくる。何も押されていない場合は 0
-----	-------------------------------------

ここでは左右のカーソルキーの入力を読み取ります。左矢印キー「←」なら 28、右矢印キー「→」なら 29 になります。

この INKEY 関数の値を変数 K へ入れて、K の値を使ってカーソルの座標 C を計算します。

$$C = C - (K = 28) * (C > 0) + (K = 29) * (C < 3)$$

キーが押されていない／押されている場合に分けて表にして、C がどんな値になるか考えてみます。まず前半の左矢印キーのチェック部分です。

キー入力	C の値	C = C -	(K = 28)	*	(C > 0)	結果
なし			0		x	C = C - 0 * x = C
←	C > 0		1		1	C = C - 1 * 1 = C - 1
←	C = 0		1		0	C = C - 1 * 0 = C

キーが押されていない場合は、C=C となり、移動しません。

左矢印キー「←」が押されている、かつ、C > 0 ならば、C = C - 1 となり、左どなりへ移動します。

左矢印キー「←」が押されている、かつ、C = 0 ならば、C = C となり、移動しません。

(C = 0 ということは、カーソルが一番左のカードにあるので、これ以上左へは移動しません)

次に後半の右矢印キーのチェック部分です。

キー入力	C の値	C = C +	(K = 29)	*	(C < 3)	結果
なし			0		x	C = C - 0 * x = C
→	C < 3		1		1	C = C + 1 * 1 = C + 1
→	C = 3		1		0	C = C + 1 * 0 = C

キーが押されていない場合は、C=C となり、移動しません。

右矢印キー「→」が押されている、かつ、C < 3 ならば、C = C + 1 となり、右どなりへ移動します。

右矢印キー「→」が押されている、かつ、C = 3 ならば、C = C となり、移動しません。

(C = 3 ということは、カーソルが一番右のカードにあるので、これ以上右へは移動しません)

このように計算式を組み合わせることで、短いプログラムでカーソルを移動できます。

```
1140 IF K!=32 GOTO 1100  
1150 BEEP:LC X+C,Y+2:" ";:RTN
```

1140 行では、K が 32 (スペースキー) でなければ、サブルーチンの先頭へもどります。
スペースキーが押されていれば 1150 行へ移り、カードを決定します。
BEEP 音を鳴らしカーソルを消して、RTN (リターン) 命令でメインプログラムへもどります。

●COM プレイヤーの思考ルーチン

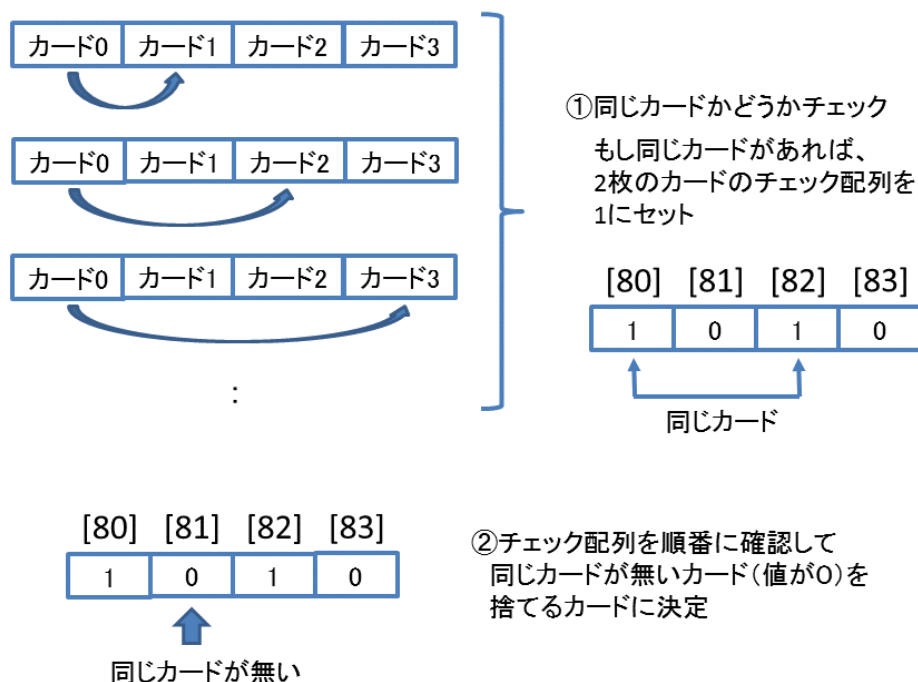
自分以外の COM プレイヤーは、乱数で適当にカードを捨てています。
これではゲームにならないので、捨てるカードをちゃんと考えて選ぶようにしましょう。

1200 ' @COMPLAY	
1210 FOR E=0 TO 2	カード 1 枚目から 3 枚目まで E をくり返し
1220 FOR F=E+1 TO 3	E の次のカードから 4 枚目まで F をくり返し
1230 IF [Q+F]=[Q+E] [80+E]=1:[80+F]=1	
1240 NEXT: NEXT	E 番目と F 番目のカードが同じだったら チェック配列変数を 1 にする
1250 FOR E=0 TO 3	チェック配列変数を確認
1260 IF [80+E]=0 C=E	他に同じカードがないカードを捨てるカードにする
1270 [80+E]=0: NEXT	次のためにチェック配列をクリア
1280 RTN	もどる

プログラムを「**RUN**」で実行します。
COM プレイヤーもちゃんと考えて、他に同じカードが無いカードを捨てるようになります。
結果として、同じカードをそろえていきます。



COM の思考ルーチンでは、配列変数[80]～[83]をチェック領域に使って、「他に同じカードがあるかどうか」をチェックして、捨てるカードを決めます。



●あがりのチェック

4 枚同じカードがそろったら、「あがり」でゲームが終了するようにしましょう。

```

200 ' @GAMELOOP
210 Q=50+P*5:D=RND(M):[Q+3]=[D]:[D]=0
220 GSB 300
230 IF [Q]=[Q+1] AND [Q]=[Q+2] AND [Q]=[
4 枚のカードが同じだったら「あがり」へジャンプ
Q+3] GOTO 290
240 Z=90+P*2:X=[Z]:Y=[Z+1]:C=3
250 IF P=0 LRUN 1,100 ELSE LRUN 1,200
260 [D]=[Q+C]:FOR N=C TO 3:[Q+N]=[Q+N+1]:NEXT:GSB 300
280 WAIT 60:P=(P+1)%4:GOTO 200
290 BEEP 10,30 BEEP 音を鳴らす
292 LC 0,10:IF P=0 ?"*YOU"; ELSE ?"*COM"
;P; あがったプレイヤーの名前を表示
294 ?" WIN*":END ゲーム終了

```

「RUN」でプログラムを実行してみましょう。
誰かのカードが 4 枚そろって「あがり」になって、ゲームが終了します。



230 行では、AND(アンド)で3つの条件式をつなげて、「4 枚のカードが同じかどうか」をチェックしています。

290 行からは「あがり」の処理です。

292 行では、プレイヤー変数 P の値によって表示を変えています。

P=0 だったら自分(YOU)があがりなので「*YOU」、それ以外だったら COM プレイヤーがあがりなので「*COM1」などとプレイヤー名を表示します。

「SAVE 0」でプログラムを保存しておきましょう。

●イチゴカードの処理

イチゴカードを引いた場合、そのプレイヤーの手札と山札を入れ替えるプログラムを追加します。

```
200 / @GAMELOOP
210 Q=50+P*5:D=RND(M):[Q+3]=[D]:[D]=0
220 GSB 300
225 IF [Q+3]=255 GSB 1300
230 GSB 300:IF [Q]=[Q+1] AND [Q]=[Q+2] A
ND [Q]=[Q+3] GOTO 290
```

もしイチゴカードを引いたら、サブルーチンと呼ぶ

カード再表示

イチゴカードの処理をするプログラムを追加します。

```
1300 / @ICHIGO
1310 IF P=0 BEEP 30,10:WAIT 60
1320 FOR C=0 TO 3
1330 E=RND(M):F=[E]
1340 IF F=255 OR F=0 GOTO 330
1350 [E]=[Q+C]:[Q+C]=F
1360 NEXT:RTN
```

イチゴカードサブルーチン

もしプレイヤーが自分だったら
BEEP 音を鳴らして1秒待つ

カードの1～4枚目でループ

山札から乱数で1枚引く

引いたカードがイチゴカードか
0(空き)だったら引き直し

手札と山札を入れ替える

くり返し:もどる

「**RUN**」でプログラムを実行してみましょう。

誰かがイチゴカードを引くと、手札が山札と入れ替わってバラバラになってしまいます。

1320～1360 行で、手札と山札を入れ替える処理をしています。

1330 行で、山札から乱数でカードを1枚選んで引きます。

1340 行では、もし引いたカードが255(イチゴカード)か0(既に引かれて空きになっているカード)だったら、1330 行にもどって選び直します。

1350 行で、選んだ山札と手札を入れ替えています。

●COM プレイヤーのカードをかくす

これで一通りゲームがプレイできることが確認できました。
最後に、COM プレイヤーのカードをかくして、見えなくします。
カード表示サブルーチンを書きかえます。

```

300 ' @PRTCARD
310 Z=90+P*2:X=[Z]:Y=[Z+1]
320 LC X,Y:IF P=0?"YOU" ELSE?"COM";P
330 LC X,Y+1
340 FOR N=0 TO 3
350 F=[Q+N]
360 IF P=G OR F=0?CHR$(F); ELSE?CHR$(2
19);
370 NEXT:RTN

```

もし自分だったら、またはカードが空きだったら、カードを表示、
COM だったら「ロ」を表示

「**RUN**」で実行してみましょう。

COM プレイヤーのカードが「ロ」になって見えなくなります。

(カタカナの「ロ」ですが、IchigoJam のフォントだと
うらがえしたカードのように見えます)



360 行で、状況によってカードの表示を変えています。

```

360 IF P=G OR F=0?CHR$(F); ELSE?CHR$(2
19);

```

IF 命令の 1 番目の条件「**P=G**」で、新たにゲーム管理変数 G を使っています。
プログラムの最初に「**CLV**」で変数をクリアしているので、G の値は通常は「0」です。
よって「P=G」は「P=0」になるので、自分 (YOU, P=0) の時に条件が成り立って、カードの表が
表示されます。

2 番目の条件「**F=0**」は、カードが空白の場合に成り立ちます。これによって、自分や
COM プレイヤーの 4 枚目のカードが無い時に、ちゃんと空白が表示されます。

さらに、「あがり」でゲームが終了する時に、4枚そろったプレイヤーのカードを表で表示するようにしましょう。

あがったプレイヤー番号を G にセットして、カードを表示

```
290 BEEP 10,30:G=P:GSB 300
292 LC 0,10:IF P=0?"*YOU";ELSE?"*COM"
;P;
294?"WIN*":END
```

「**RUN**」でプログラムを実行してみましょう。
COM プレイヤーがあがった時に、カードが表で表示されます。



290 行で、ゲーム管理変数 G にその時のプレイヤー番号 P をセットして、カード表示サブルーチン呼びます。
これであがったプレイヤーのカードが表示されるようになります。

★できる人は

今回作った「イチゴトランプ」は、マンガ「アタゴオル物語」(ますむらひろし作)に出てくる「ネズミトランプ」を元にしています。

カードの種類を変えたり、イチゴカードのような特殊カードを追加したりして、改造すると面白いでしょう。

また、同様の作り方で、通常のトランプゲームや UNO などでも作れるでしょう。
トランプ(カード)ゲームでは、COM の思考ルーチンをどう作るか? が一番のポイントで、それによってコンピュータが強くなったり弱くなったりして、ゲームの面白さが変わります。
いろいろなゲームを作ってみてください。