IchigoJam BASIC リファレンス ver 1.4

コマンド	解説	例
LED / エル・イー・デ		
イー	数が1ならLEDが光り、0なら消える	LED 1
<u> </u>	│ │数1の数だけ待つ(60でl秒)。マイナスの数を指定すると走査線分で待つ(-261でWAIT1と同じ時間)。省略できる数2に0を指定すると画面表示を	
WAIT / ウェイト	·	WAIT 60
	止め低電力化して待つ。	WAIT OO LED 4
:/コロン	コマンドを後ろに続けて書くときの区切り記号	WAIT 60:LED 1
1/ワン	1. 数は-32768から32767まで表記できる。 2. 行頭の1~32767で指定された数は行番号として、コマンドと合わせて記録する。コマンドを省略する	10 LED1
	とその行を削除(16385以上は指定しないと表示しない)	
RUN / ラン	プログラムを実行する [F5]	RUN
LIOT / II 7 l	プログラムを表示する [F4] (行番号1で1行表示、行番号1がマイナスでその行まで表示、行番号2指定でその行まで表示、行番号2が0の時終わりま	LIST 10 200
LIST / リスト	, 「で表示、ESCで途中停止)	LIST 10,300
GOTO / ゴートゥー	指定した行番号へ飛ぶ(式も指定可能)	GOTO 10
END/エンド	プログラムを終了する	END
IF/イフ	ンロックなとNCT する 数が0でなければコマンド1を実行し、0であればコマンド2を実行する (THEN,ELSE以降は省略可)	IF BTN() END
THEN / ゼン	数が0でなければコマンド1を実行し、0であればコマンド2を実行する (THEN,ELSE以降は省略可)	IF BTN() THEN END
ELSE / エルス	数が0でなければコマンド1を実行し、0であればコマンド2を実行する (THEN,ELSE以降は省略可)	IF BTN() END ELSE CONT
BTN / ボタン	ボタンが押されていれば1、そうでないとき0を返す(数:0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE/X(88)、省略で0、−1でビットパターンで返	LED BTN()
	す)	LEB BTN()
NEW / = ¬ -	プログラムを全部消す	NEW
PRINT / プリント	数や文字列を表示する (文字列は"で囲む、";" で連結できる)	PRINT "HI!";15
?/クエスション	数や文字列を表示する (文字列は"で囲む、";" で連結できる)	?"HI!";15
.,, _,,,	次に文字を書く位置を横、縦の順に指定する(左上が0,0、縦=-1で無表示)。数1のみでX+Y*幅。数3が0でなければ指定した場所にカーソルを表示	, ,
LOCATE / ロケート		LOCATE 3,3
	する。	
LC / ロケート	次に文字を書く位置を横、縦の順に指定する(左上が0,0、縦=-1で無表示)。数1のみでX+Y*幅。数3が0でなければ指定した場所にカーソルを表示	LC 3,3
	する。	
CLS / クリア・スクリ	 画面を全部消す	CLS
ーン	— µµ < ⊥ нг/П /	
RND / ランダム	0から数未満のでたらめな整数を返す	PRINT RND(6)
SAVE / セーブ	プログラムを保存する(0~3の4つ、100-227 外付けEEPROM、省略で前回使用した数) ボタンを押した状態で起動すると0番を読み込み自動実行	SAVE 1
LOAD/ロード	プログラムを読み出す(0~3の4つ、100-227 外付けEEPROM、省略で前回使用した数)	LOAD
FILES / ファイルズ	数1(省略可)〜数2のプログラム一覧を表示する(EEPROM内ファイル表示に対応、0指定ですべて表示、ESCで途中停止)	FILES
BEEP / ビープ	単音を鳴らす 周期(0-32767)と長さ(1/60秒単位)は省略可 *SOUND(EX2)-GNDに圧電サウンダーなどの接続が必要	BEEP
	MML、記述した音楽を鳴らす。省略で音停止 *SOUND(EX2)-GNDに圧電サウンダーなどの接続が必要 (MML、CDEFGAB:ドレミファソラシ、	
PLAY / プレイ	R:休符、.:音や休符を1.5倍伸ばす、T120:テンポ、O4:オクターブ(1-6)、N10:単音、<:オクターブ上げる、>:オクターブ下げる、\$:繰り	PLAY "\$CDE2CDE2"
	返し、': 以後鳴らさない)	
TEMPO / テンポ	再生中の音楽のテンポを変更する	TEMPO 1200
+ / プラス	足し算する	PRINT 1+1
- / マイナス	1. 引き算する 2. 後に続く数をマイナスにする	PRINT 2-1
*/アスタリスク	掛け算する	PRINT 7*8
// スラッシュ	割り算する(小数点以下は切り捨て)	PRINT 9/3
% / パーセント	割り算した余りを返す	PRINT 10%3
() / ブラケット	カッコ内の式を優先して計算する	PRINT 1+(1*2)
LET / レット	数をアルファベット 1 文字の変数や配列の値として記憶する(配列へ連続代入可能 LET[0],1,2)	LET A,1
 11		
=/イコール	3. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる	A=1
=/イコール	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる	
=/イコール INPUT/インプット	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可)	A=1 INPUT "ANS?",A
= / イコール INPUT / インプット TICK / ティック	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間	A=1
= / イコール INPUT / インプット TICK / ティック CLT / クリア・ティッ	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可)	A=1 INPUT "ANS?",A
= / イコール INPUT / インプット TICK / ティック CLT / クリア・ティッ ク	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする	A=1 INPUT "ANS?",A PRINT TICK() CLT
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY()
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト RIGHT/ライト	 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト RIGHT/ライト	 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト RIGHT/ライト UP/アップ	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト RIGHT/ライト UP/アップ DOWN/ダウン SPACE/スペース	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65)
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A")
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72に対する文字で返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2
=/イコール INPUT/インプット TICK/ティック CLT/クリア・ティック INKEY/インキー LEFT/レフト RIGHT/ライト UP/アップ DOWN/ダウン SPACE/スペース CHR/キャラクター ASC/アスキー SCROLL/スクロール SCR/スクリーン	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0)
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72を返す。INKEYのキーコードの判定や、SCROLLに使う 72に対する文字で返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 13定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0)
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0)
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 曲面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す。	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 13定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0)
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル !=/ ノット・イコー	1. 数をアルファベット 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードの対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラかスクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地較して等しい時に1、それ以外で0を返す。 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A!=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル !=/ ノット・イコール・トゥー ◇ / レス・アンド・グ	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む) *数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 曲面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す。	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール != / ノット・イコール・トゥー ◇ / レス・アンド・グレーター・ザン	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードの判定や、SCROLLに使う PRINT内で、文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\sigma B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル・トゥー ◇ / レス・アンド・グレーター・ザン <= / レス・ザン・オ	1. 数をアルファベット 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードの対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラかスクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地較して等しい時に1、それ以外で0を返す。 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A!=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル・トゥー ◇ / レス・アンド・グレーター・ザン <= / レス・ザン・オア・イコール・トゥー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 曲面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<>B LED 1 IF A<=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ◇ / レス・アンド・グレーター・ザン <= / レス・ザン・オア・イコール・トゥー / レス・ザン</td <td>1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードの判定や、SCROLLに使う PRINT内で、文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す</td> <td>A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\sigma B LED 1</td>	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードの判定や、SCROLLに使う PRINT内で、文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\sigma B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール != / ノット・イコール・トゥー ◇ / レス・アンド・グレーター・ザン <= / レス・ザン・オア・イコール・トゥー / レス・ザン = / グレーター・ザ	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<>B LED 1 IF A<>B LED 1 IF A <b 1<="" led="" td="">
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル !=/ ノット・イコール・トゥー ペ / レス・デンド・グレーター・ザン <=/ レス・ザン・オア・イコール・オア・イコール・	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 曲面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<>B LED 1 IF A<=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール != / ノット・イコール・トゥー ◇ / レス・アンド・グレーター・ザン <= / レス・ザン・オア・イコール・トゥー / レス・ザン = / グレーター・ザ	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<>B LED 1 IF A<>B LED 1 IF A <b 1<="" led="" td="">
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル !=/ ノット・イコール・トゥー ペ / レス・デンド・グレーター・ザン <=/ レス・ザン・オア・イコール・オア・イコール・	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す 比較して以下の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<>B LED 1 IF A<>B LED 1 IF A <b 1<="" led="" td="">
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ペ / レス・アンド・グレーター・ザン < / レス・ザン >= / グレーター・ザン・オア・イコール・トゥー < / レス・ザン >= / グレーター・ザン・オア・イコール・トゥー	1. 数をアルファベット 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 92を返す。INKEYのキーコードの判定や、SCROLLに使う 97をINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(O/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較して未満の時に1、それ以外で0を返す 比較して未満の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ◇ / レス・アンド・グレーター・ザン < / レス・ザン >= / グレーター・ザン > / グレーター・ザン > / グレーター・ザン	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(O/UP-L、1/RIGHT名、2/DOWN:下、3/LEFT左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 曲面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す 比較して以上の時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール ル・トゥー ◇ / レス・アンド・グレーター・ザン <= / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン AND / アンド	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARITからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARITから 1 文字入力する(入力がない時は0、UARITからが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 20を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードと対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(0/UP:上、1/RIGHT右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 地転して等した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す と乾してより大きい時に1、それ以外で0を返す と乾してより大きい時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ◇ / レス・アンド・グレーター・ザン <ー/レス・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン AND / アンド && / アンド OR / オア	1. 数をアルファベット1文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)・数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから1文字入力する(入力がない時は、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの利定や、SCROLLに使う 29を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 12定した方向に1キャラクスクロールする(0/UP・上、1/RIGHT-5、2/DOWN:下、3/LEFT-左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等したの間に、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す 比較してより大きい時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\(-\text{B}\) LED 1 IF A\(-\text{B}\) LED 1 IF A\(-\text{B}\) LED 1 IF A>B LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー	1. 数をアルファベット1文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 数を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返すこれに使う 12を返することに使う 12を返するといて使う 12を返することに使う 12を返することに使う 12を返することに使う 12を返するといて使う 12を返することに使う 12を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(のUP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す。 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較してすい時に1、それ以外で0を返す 比較して以上の時に1、それ以外で0を返す ともの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式が1の時に1、それ以外で0を返す とちらの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー	1. 数をアルファベット1文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60°261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから1 文字入力する(入力がない時は0、UARTから0が入力された時は#100) 28を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 数を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返す。INKEYのキーコードの利定や、SCROLLに使う 12を返すこれに使う 12を返することに使う 12を返するといて使う 12を返することに使う 12を返することに使う 12を返することに使う 12を返するといて使う 12を返することに使う 12を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略可) CLTからの時間を返す(1/60秒で1進む)*数に1指定で1/(60*261)秒で1進む時間 TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 30を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う 32を返す。INKEYのキーコードの利定や、SCROLLに使う PRINT内で、文字コードに対応する文字を返す(コンマ区切りで連続表記可) 文字に対する文字コードを返す 指定した方向に1キャラ分スクロールする(のUP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左) 画面上の指定した位置に書かれた文字コードを返す(指定なしで現在位置) 比較して等しい時に1、それ以外で0を返す。 比較して等しい時に1、それ以外で0を返す 比較して等しくない時に1、それ以外で0を返す 比較してすい時に1、それ以外で0を返す 比較して以上の時に1、それ以外で0を返す ともの式も1の時に1、それ以外で0を返す とちらの式も1の時に1、それ以外で0を返す とちらの式が1の時に1、それ以外で0を返す とちらの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す とちらかの式が1の時に1、それ以外で0を返す	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ペ / レス・アンド・グレーター・ザン <= / レス・ザン >= / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン・オア・イコール・トゥー > / グレーター・ザン AND / アンド && / アンド OR / オア II / オア NOT / ノット ! / ノット ! / ノット	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる オーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは貨幣可) TICK(のカウントをリセットする オーボードやUARTから 1 文字入力する(入力がない時は0、UARTからが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INMEYのキーコードの判定や、SCROLLに使う 32を返す。INMEYの中でを返す 上飲して対するの実力に、それ以外でのを返す とい方のの時に1、それ以外でのを返す とちらかの式が1の時に1、それ以外でのを返す とちらかの式が1の時に1、それ以外でのを返す とちらかの式が1の時に1、それ以外でのを返す とちらかの式が1の時に1、それ以外でのを返す とちらかの式が1の時に1、それ以外でのを返す これ以降のでを返す これ以降のできを実行しない(コメント機能)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1 IF NOT A=1 LED 1 IF I(A=1) LED 1
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ル・トゥー ペ / レス・アンド・グレーター・ザン <= / レス・ザン >= / グレーター・ザン ン・オア・イコール・トゥー > / グレーター・ザン AND / アンド OR / オア II / オア NOT / ノット REM / リマーク ' ゾシングル・クォート	 1. 数をアルファベット 1 文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で破値を要数にいれる(文字列とコンマは貨幣可) (CLTからの時間を返す(1/6の秒で1進む)) 数に1指定で1/(60*261)秒で1進む時間 (TICK()のカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時は0. UARTからが入力された時は#100) 28を返す。INKEYのキーコードの判定や、SCROLLに使う 29を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードの判定や、SCROLLに使う 22を返す。INKEYのキーコードを返す (却定なして現在位置) 立即上の指定した位置に書かれた文字コードを返す (指定なして現在位置) 立即上の指定した位置に書かれた文字コードを返す (指定なして現在位置) 立即上の指定した位置に書かれた文字コードを返す (指定なして現在位置) 上較して等しくない時に1、それ以外で0を返す 上較して等しくない時に1、それ以外で0を返す 上較して以上の時に1、それ以外で0を返す とちらののまた1、それ以外で0を返す とちらののまた1の時に1、それ以外で0を返す とちらののまた1の時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりの時に1、それ以外で0を返す こだりののまで2×ドを実行しない (コメント機能) 	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1 IF II B=1 LED 1 IF II REM START START
= / イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字別とコンマは省略可) 「ICK(ののカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時はの、UARTからが入力された時は#100) 20を返す。INKEYのキーコードの判定や、SCROLLに使う 20を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 30を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 31を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYのキーコードの判定や、SCROLLに使う 32を返す。INKEYの手に、それ以外でのを返す と比較してよるい時に1、それ以外でのを返す とちらの式が1の時に1、それ以外でのを返す とちらので式が1の時に1、それ以外でのを返す とちらのの式が1の時に1、それ以外でのを返す とちらのの式が1の時に1、それ以外でのを返す これ以降のコマンドを実行しない(コメント機能) 変数に数1をいれ、数2になるまで数3ずつ場やしながらNEXTまでをくりかえす(STEPは省略可、6段まで)	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A<=B LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 II B=1 LED 1 IF NOT A=1 LED 1 IF I(A=1) LED 1 REM START START FOR I=0 TO 10:?I:NEXT
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ペ / レス・デンド・グレーター・ザン・オア・イコール・トゥー レス・ザン = / グレーター・ザン・オア・イコール・トゥー > / グレーター・・ザン・オア・イコール・トゥー > / グレーター・・ザン・オア・ゲークー・・サンドの () アンド && / アンド CR / オア II / オア NOT / ノット REM / リマーク ' / シングル・クォート FOR / フォー TO / トゥー	 1、数をアルファベット)文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略同) TICK(ののカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時はの、UARTからのが入力された時は#100) 28を返す。INKEYのキーコートの利定や、SCROLLに使う 28を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 22を返す。INKEYのキーコートの利定や、SCROLLに使う 32を返す。INKEYのキーコートの利定や、SCROLLに使う 32を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートで必可定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートで必可定や。 SCROLLに使う 30を返すメラコードを返す 用途した位置に書かれた文字コートを返す(日常なして現在位置) 動面上の指定した位置に書かれた文字コートを返す(指定なして現在位置) 地較して等しい的に、セイ以外でのを返す 上較して等しい的に1、それ以外でのを返す 上較して等しい的に1、それ以外でのを返す 上較して以上の時に1、それ以外でのを返す 上較して以上の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す ごちののがに1、それ以外でのを返す ごがのの時に1、それ以外でのを返す これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) 	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\B LED 1 IF A\B LED 1 IF A\B LED 1 IF A=1 AND B=1 LED 1 IF A=1 AND B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1 IF I(A=1) LED 1 REM START START FOR I=0 TO 10:?I:NEXT
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール・トゥー ペ / レス・ザン・オア・イコール・トゥー レス・ザン = / グレーター・ザン AND / アンド OR / オア II / オア NOT / ノット FOR / フォー TO / トゥー STEP / ステップ	1. 数をアルファベット1文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは増積可) 【ICK()のカウントをリセットする キーボードやUARTから1文字入力する(入力がない時は0、UARTからのが入力された時は#100) 28を返す。INKEYのキーコードの制定や、SCROLLに使う 20を返す。INKEYのキーコードの制定や、SCROLLに使う 30を返す。INKEYのキーコードの制定や、SCROLLに使う 30を返す INKEYのキーコードの制定や、SCROLLに使う 30を返す INKEYのキーコードの制定や、SCROLに使う 30を返す INKEYのキーコードの制定や、SCROLLに使う 30を返す INKEYのキーコードの制定や、SCROLに使う 30を返す INKEYのキーロードの制定や、SCROLに使う 30を返す INKEYのキーロードの制定や、SCROLに使う 30を返す INKEYのキーロードの制定や、SCROLに使う 30を返す INKEYのキーロードの制定や、SCROLに使う 30を返す INKEYのキーロードの制定や、SCROLに使う 30を図が、INKEYのキーロードの制定や、SCROLに使う 30を図が、INKEYのキーロードの制定や、SCROLに使う 30を図が、INKEYのキーロードの制定や、SCROLに使う 30を図が、INKEYのキーロードの制定や、SCROLに使う 30を図が、INKEYのキーロードの制定をはないます 30を図が、INKEYのキーロードの制定をはないます 30を図が、INKEYのキーロードの制定をはないます 30を図が、INKEYのキーロードの制定をはないます 30を図が、INKEYのものではないます 30を図が、INKEYのキーロードの制定をはないます 30を図が、INKEYのものではないます 30を図が、INKEYのものではないないます 30を図が、INKEYのものではないます 30を図が、INKEYのものではないます 30を図が、INKEYのものではないます 30を	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=UP LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\Rightarrow B LED 1 IF A\Rightarrow B LED 1 IF A\Rightarrow B LED 1 IF A=1 AND B=1 LED 1 IF A=1 AND B=1 LED 1 IF A=1 OR B=1 LED 1 IF A=1 IF B LED 1 IF A=1 IF A=1 LED 1 IF A=1 IF
=/イコール INPUT / インプット TICK / ティック CLT / クリア・ティック INKEY / インキー LEFT / レフト RIGHT / ライト UP / アップ DOWN / ダウン SPACE / スペース CHR / キャラクター ASC / アスキー SCROLL / スクロール SCR / スクリーン VPEEK / ブイ・ピーク == / イコール・イコール != / ノット・イコール・トゥー ◇ / レス・ザン・オア・イコール・トゥー レス・ザン = / グレーター・ザン AND / アンド OR / オア II / オア NOT / ノット FOR / フォー TO / トゥー	 1、数をアルファベット)文字の変数や配列の値として記憶する。2. 式として使うと==と同じ機能となる キーボードやUARTからの入力で数値を変数にいれる(文字列とコンマは省略同) TICK(ののカウントをリセットする キーボードやUARTから 1 文字入力する(入力がない時はの、UARTからのが入力された時は#100) 28を返す。INKEYのキーコートの利定や、SCROLLに使う 28を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 31を返す。INKEYのキーコートの利定や、SCROLLに使う 22を返す。INKEYのキーコートの利定や、SCROLLに使う 32を返す。INKEYのキーコートの利定や、SCROLLに使う 32を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートで必可定や、SCROLLに使う 30を返す。INKEYのキーコートの利定や、SCROLLに使う 30を返す。INKEYのキーコートで必可定や。 SCROLLに使う 30を返すメラコードを返す 用途した位置に書かれた文字コートを返す(日常なして現在位置) 動面上の指定した位置に書かれた文字コートを返す(指定なして現在位置) 地較して等しい的に、セイ以外でのを返す 上較して等しい的に1、それ以外でのを返す 上較して等しい的に1、それ以外でのを返す 上較して以上の時に1、それ以外でのを返す 上較して以上の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す どちらの式も1の時に1、それ以外でのを返す ごちののがに1、それ以外でのを返す ごがのの時に1、それ以外でのを返す これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) これ以降のコマンドを実行しない(コメント機能) 	A=1 INPUT "ANS?",A PRINT TICK() CLT PRINT INKEY() IF INKEY()=LEFT LED1 IF INKEY()=RIGHT LED1 IF INKEY()=DOWN LED1 IF INKEY()=DOWN LED1 IF INKEY()=SPACE LED1 PRINT CHR\$(65) PRINT ASC("A") SCROLL 2 PRINT SCR(0,0) PRINT VPEEK(0,0) IF A==B LED 1 IF A\B LED 1 IF A\B LED 1 IF A\B LED 1 IF A=1 AND B=1 LED 1 IF A=1 AND B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1 IF A=1 II B=1 LED 1 IF I(A=1) LED 1 REM START START FOR I=0 TO 10:?I:NEXT

POS / ポジション	カーソル位置を返す(数、省略時または0:X+Y*幅、1:X座標、2:Y座標) *ver1.4以上	?POS(0),POS(1)
DRAW/ドロー	数1,数2の座標から数3,数4の座標へ線を引く(座標は最大63x47)、数5に0指定で線を消し、2指定で反転する、省略時または1指定で線を引く *ver1.4以上	DRAW 1,5,10,15
POINT / ポイント	数1,数2の座標にDRAWで描かれた点または文字があるときに1、そうでないとき0を返す *ver1.4以上	?POINT(1,5)
OUT / アウト	外部出力OUT1-11に0または1を出力する 数2を省略でまとめて出力できる(数2に-1指定でINへ切り替え、-2指定でプルアップ付きINへ切り替え ※IN3は除く)	OUT 1,1
IN/イン	INO-10から入力する(0または1) 数を省略してまとめて入力できる (INO,1,4,9はプルアップ、IN5-8,10-11はOUTで切り替え時使用可能、INO,9はボタン)	LET A,IN(1)
ANA / アナログ	外部入力の電圧(0V-3.3V)を0-1023の数値で返す(2:IN2、5-8:IN5-8(OUT1-4)、0,9:BTN、省略で0)	?ANA()
PWM / ピー・ダブリ ュー・エム	外部出力OUT2-5に数2で0.01msec単位で指定するパルスを出力する(0-2000、周期20msec)、数3で周期を指定(省略時2000=20msec、マイナス 値指定で周期1/480)	PWM 2,100
CLV / クリア・バリア ブル	変数、配列を全部0にする	CLV
CLEAR / クリア		CLEAR
CLK / クリア・キー	キーバッファとキーの状態をクリアする	CLK
CLO / クリア・アウト プット	入出力ピンを初期状態に戻す	CLO
ABS / アブソリュート [] / アレイ	絶対値を返す(マイナスはプラスになる) 配列([0]から[101]までの102コの連続した変数として使える) LET[0],1,2,3で連続代入可能	?ABS(-2) [3]=1
GOSUB / ゴーサブ	戻り先をスタックに積み、数または式で指定した行番号に飛ぶ(ネストは30段まで)	GOSUB 100
GSB / ゴーサブ	戻り先をスタックに積み、数または式で指定した行番号に飛ぶ(ネストは30段まで)	GSB 100
RETURN / リターン RTN / リターン	GOSUB/GSBの呼び出し元へ戻る GOSUB/GSBの呼び出し元へ戻る	RETURN RTN
DEC / デシ	PRINT内で、数を文字列にする(2番目の数は桁数、省略可)	?DEC\$(99,3)
#/ハッシュ	16進数で数を表記する *この後THENを省略しA~Fで始まるコマンドを使う場合:で区切る	#FF
HEX / ヘックス	PRINT内で、数を16進数の文字列にする(2番目の数は桁数、省略可)	?HEX\$(255,2)
` / バック・クォート BIN / バイナリー	2進数で数を表記する PRINT内で、数を2進数の文字列にする(2番目の数は桁数、省略可)	`1010 ?BIN\$(255,8)
& / アンパサンド	PRINT内で、数を2進数の大子列にする(2番目の数は削数、自略可) 論理積(ビット演算)	?3&1
1/パイプ	論理和(ビット演算)	?3l1
^/八ット	排他的論理和(ビット演算)	?A^1
>>/シフト・ライト	右シフトする(ビット演算) 左シフトする(ビット演算)	?A>>1 ?A<<1
~/ チルダ	ビット反転 (ビット演算)	?~A
COS/コサイン	指定された数を角度の度数としてコサインの値の256倍を返す *ver1.4以上	?COS(90)
SIN / サイン	指定された数を角度の度数としてサインの値の256倍を返す *ver1.4以上	?SIN(90)
STOP / ストップ	プログラムを中断する	STOP
CONT / コンティニュ ー	実行中の行や、中断した行を再度実行する	CONT
SOUND / サウンド	音が再生中なら1、そうで無いとき0をを返す	?SOUND()
EDEE / DIII		OFDEE()
FREE / フリー	プログラムの残りメモリ数を返す Ichigo lam BASICのバージョン番号を返す	?FREE()
FREE / フリー VER / バージョン RENUM / リナンバー	IchigoJam BASICのバージョン番号を返す	?FREE() ?VER() RENUM
VER / バージョン		?VER()
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す	?VER() RENUM LRUN 1 ?FILE()
VER / バージョン RENUM / リナンバー LRUN / ロードラン	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0)	?VER() RENUM LRUN 1 ?FILE() ?LINE()
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリネップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC"
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリミシカーパターンの#7FFなど) メモリ・の書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー 数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC")
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)*前方一致 画面表示非表示を切り替える 0で画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEO0時CPUクロックを1/数2に変更し省電力化、	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)*前方一致 画面表示非表示を切り替える 0で画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEO0時CPUクロックを1/数2に変更し省電力化、	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット	IchigoJam BASICのパージョン番号を返す	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エ	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み (連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)※前方一致 画面表示非表示を切り替えるので画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1が2の倍数で白黒反転、数1が3以上で拡大モード IchigoJamをリセットする プログラムを休止する(ボタンを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定(0:オフ、1:PRINTのみ、2:PRINTLC/CLS/SCROLL、3:PRINTのみ/改行コードVn、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2:シリアル受信設定(0:オフ、1:75600bps、-2:38400bps、-100以下指定で指定した数の-100倍bpsに指	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダ ム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート	IchigoJam BASICのバージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み、書き込み行った数を返す 現在実行中の行番号を返す(非実行時はの) 種を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリマップを表示する メモリスの書き込み (運輸書き込み可能 POKE#700,1,2,3) メモリニー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す (数2(省略可)で長さ指定) 文字列の長さを返す 行の先頭に書くとラベルとなり、行番号の代わりとして使える (GOTO @LOOPなど) ※前方一数 画面表示非表示を切り替える 0で画面表示を停止し処理高速化 (F8で表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1が2の倍数で自黒皮底、数1が3以上で拡大モード IchigoJamをリセットする プログラムを休止する (ボタンを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定 (0:オフ、1:PRINTのみ、2:PRINTのよので行っていれ、4-4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2:シリアル受信設定 (0:オフ、1:オン、+2:ESC無効、+4:CR変換(13→10) 省略時1) シリアル過信速度を変更する(0:マ初期値の115,200bps、-1:57600bps、2:38400bps、-100以下指定で指定した数の-100倍bpsに指定、-2304:230400bps)。数2で12Cの通信速度設定(単位kHz、0:デフォルト400kHz)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エス OK / オーケー 2CR / アイ・ツー・シ	tchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み後、実行時は0) 種を指定して乱数を初期化する メモリ読み出し(キャラクターパターンの#7FFなど) メモリへの書き込み (連続書き込み可能 POKE#700,1,2,3) メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 「つの先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 「つの先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)※前方一致 画面表示表示を切り替える 0で画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1が2の倍数で白黒反転、数1が3以上で拡大モード lchigoJamをリセットする プログラムを休止する(ボタンを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定(0:オフ、1:PRINTのみ、2:PRINTLC/CICI/S/CROLL、3:PRINTのみのだつコードVYn、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2:シリアル受信設定(0:オフ、1:オン、+2:ESC無効、+4:CR変換(13→10) 省略時1) シリアル通信速度を変更する(0で初期値の115;200bps、-1:57600bps、-2:38400bps、-100以下指定で指定した数の-100倍bpsに指定、-2304:230400bps)。数2で12Cの通信速度設定(単位水14、0:デフォル+400kHz) D(Kやエラーメッセージの表示有無を切り替える(数、1:表示、2:非表示、省略で1)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エス OK / オーケー !2CR / アイ・ツー・シー・リード	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み後、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー 数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の完頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)**前方一致 画面表示非表示を切り替える 0で画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEO0時CPUクロックを1/数2に変更し省電力化、数1か2の倍数で自規反転、数1が3以上で拡大モード IchigoJamをリセットする プログラムを休止する (ポタンを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定(C3・オフ、1: PPINTのみ、2: PRINT/LC/CLS/SCROLL、3: PRINTのみ/成行コードV/n、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2: シリアル通信波を変更する(0で初別値 0115, 2000pos、+1: 57600bps、+2: 28400bps、+1: 0以下指定で指定した数の-100倍bpsに指定、2304: 2304: 2304: 2304: 2000pps)。数2 で12Cの通信速度設定(単位kHz、0: デフォルト400kHz) OKやエラーメッセージの表示有無を切り替える(数、1表示、2:非表示、省略で1) I2Cで周辺機器に書き込む I2Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ(如2V下送信が1byteの時数3を省略可) I2Cで周辺機器に書き込む I2Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ(数4/数5は省略可、コマンド送信が1byteの時数3を省	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パター ン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エス OK / オーケー 12CR / アイ・ツー・シー・リード I2CW / アイ・ツー・シー・ライト IOT.IN / アイ・オー・	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み後、書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリミップを表示する メモリ読み出し(キャラクターパターン0-#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコピー 数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の先頭アドレスを返す 行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)※前方一致 画面表示非表示を切り替えるので画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEO0時CPUクロックを1/数2に変更し省電力化、数1が20人を依止する(ボクシを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定(の:オフ、1:PRINTOみ、2:PRINTLC/CLS/SCROLL、3:PRINTのみ/改行コードVn、+4で入力エコーパック、+8で画面表示 プログラムを休止する(ボクンを押すと起動し、LRUN0を実行する) 数1:シリアル出力設定(の:オフ、1: オン、+2:ESC無効、+4-CR変換(13・10) 省略時) シリアル通信速度を変更する(0で初期値つ115,200pps、-1:57600bps、-2:38400bps、-1:00以下指定で指定した数の-100倍psに指定、-2:304-230400bps)。数2:マizCの通信速度設定(単位内は、0:デフォルト400kHz) OKやエラーメッセージの表示有無を切り替える(数、1表示、2:非表示、省略で1) 区で両辺機能から読み込む I2Cアドレス、コマンド送信アドレス・長き、受信アドレスと長き(コマンド送信が1byteの時数3を省略可、コマンド送信が1byteの時数3を省略可、コマンド送信が1byteの時数3を省略可、コマンド送信が1byteの時数3を省略可、コマンド送信が1byteの時数2/数3を省略可	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・エス OK / オーケー 12CR / アイ・ツー・シー・リード 12CW / アイ・ツー・シー・ライト IOT.IN / アイ・オ	IchigoJam BASICのパージョン番号を返す プログラムの行番号を数付省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、実行する 最後にプログラムを読み込み。書き込み行った数を返す 現在実行中の行番号を返す(非実行時は0) 種を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリニの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコークラン・パケーンの#7FFなど) メモリへの書き込み(連続書き込み可能 POKE#700,1,2,3) メモリコーク 数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の長さを返す 「分の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)**前方一数 画面表示表示を切り替えるので画面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1が2の倍数で白黒反転、数1が3以上で拡大モード IchigoJamをリセットする プログラムを休止する「ボタンを押すと起動し、LRUNOを実行する) 数1:シリアル出力設定(0:オフ、1:PRINTのみ、2:PRINTLC/CLS/SCROLL、3:PRINTのみ/改行コードVn、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2:シリアル受信設定(0:オフ、1:オン、+2:ESC無効、+4-CR変換(13→10) 省略時1) シリアル適信速度を変更する(0:が初期値の115,200bps、-1:5760bps、-2304(230400bps)、数で12Cの通信速度設定(単位内は、0:デフォルト400kt2) OKやエラーメッセージの表示有無を切り替える(数、1:表示、2:非表示、省略で1) I2Cで周辺機器から読み込む 12Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ(数4数5は省略可、コマンド送信が1byteの時数3を省略可)	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2) R=I2CW(#50,#114A,2,#114C,2)
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エス OK / オーケー 12CR / アイ・ツー・シー・リード 12CW / アイ・ツー・シー・ライト IOT.IN / アイ・オー・ティー・イン IOT.OUT / アイ・オー・アート WS.LED / ダブリュ	IchigoJam BASICのパージョン番号を返す プログラムの行品号を数((省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 以本モリマップを表示する メモリマップを表示する メモリマップを表示する メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の先頭アドレスを返す 「行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)※前方一致 画面表示非表示を切り替えるので画面表示を停止し処理局態化(FBで表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1/52の倍数で自具を版、数/が32に上で拡大モード IchigoJamをリセットする プログラムを作止する (ボタンを押すと起動し、LRUN0を実行する) 数1: シリアル出力放定(のオフ、1: PRINTのみ、2: PRINTLO(CLS/SCROLL、3: PRINTのみの放行コードVin、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2: シリアル受信該定(の:オフ、1: オン、+2: ESC無効、+4: CR変換(13 → 10) 省略時1) シリアル通信速度を変更する(ので初期値の115,2000ps、-1: 5/000ps、-2: 38400bps、-100以下指定で指定した数の-100倍bpsに指定、-2304(2300bps)、数2で120の通信速度放変度)、単位がLRV 4400kHz) OKやエラーメッセージの表示有無を切り替える(数、1:表示、2: 非表示、省略で1) I2Cで周辺機器から読み込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(数4)数5は省略可、コマンド送信が1byteの時数3を省略可) I2Cで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(数4)数5は省略可、コマンド送信が1byteの時数3を省略可) sakura.loモジュールへ数をチャンネルので即時送信する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2) R=I0T.IN() IoT.OUT 100
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・エス OK / オーケー 12CR / アイ・ツー・シー・リード 12CW / アイ・ツー・シー・ライト IOT.IN / アイ・オー・ティー・イン IOT.OUT / アイ・オー・アウト WS.LED / ダブリュー・エス・エル・イー・ディー	chigoJam BASICのパージョン番号を返す プログラムを済み込み後、実行する 最後にプログラムを誘み込み、書き込み行った数を返す 現在実行中の行番号を数す(排実行動はの) 権を指定して乱数を初期化する メモリマップを表示する メモリマップを表示する メモリマップを表示する メモリアップを表示する メモリアップを表示する メモリアップを表示する メモリコレードレスへ数なのアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) モャラクターパターン(470-47F)を初野化する マ学列の先端アドレスを返す アドレスを返す アドレスを返す アドレスを返す アドルスを返す アルルとなり、行番号の代わりとして使える(GOTO @LOOPなど)。4前方一数 間面表示:非表示を切り替えるので請面表示を停止し処理高速化(F8で表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1が2の信頼で自患反転、数1が3以上で拡大モド にhigoJamをリセットする アログラムを休止する(ボタンを押すと起動し、LRUNのを実行する) 数1: シリアル出力設定(ルオス・1:PPINITのみ、2:PRINITのより、3:PRINITのか(次行コードがれ、44で入力エコーバック、48で画面表示 OFF、初期値2)、数2: シリアル受信設定(のオフ、1:72・1×2・1×2・1×2・2とSS無別、44・CR変換(13-11の) 省略時1) シリアル出力設定(ルオス・1:PPINITのみ、2:PRINITALCICLES/SCROLL、3:PRINITのか(次行コードがれ、44で入力エコーバック、48で画面表示 OFF、初期値2)、数2: シリアル受信設定(単位水は、0:デフォルト4000kHz) CCで周辺機器がら読み込む 12Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ(コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(カ4分5は省略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(カ4分5は省略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(カ4分5は音略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(カ4分5は音略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(カ4分5は音略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ (カ4分5は音略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、受信アドレス・長さ、受信アドレス・長さ、受信アドレスと長さ (カ4分5は音略可、コマンド送信が1byteの時数3を省略可) ICCで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、受信を表する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2) R=I0T.IN() IoT.OUT 100 WS.LED 3
VER / バージョン RENUM / リナンバー LRUN / ロードラン FILE / ファイル LINE / ライン SRND / エス・ランダム HELP / ヘルプ PEEK / ピーク POKE / ポーク COPY / コピー CLP / クリア・パターン " / ダブル・クォート STR / ストリング LEN / レングス @ / アット・マーク VIDEO / ビデオ RESET / リセット SLEEP / スリープ UART / ユー・アート BPS / ビー・ピー・エス OK / オーケー 12CR / アイ・ツー・シー・リード 12CW / アイ・ツー・シー・ライト IOT.IN / アイ・オー・ティー・イン IOT.OUT / アイ・オー・ティー・アウト WS.LED / ダブリュー・エス・エル・イ	IchigoJam BASICのパージョン番号を返す プログラムの行品号を数((省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある プログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 最後にプログラムを読み込み後、共行する 以本モリマップを表示する メモリマップを表示する メモリマップを表示する メモリコピー数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる) キャラクターパターン(#700-#7FF)を初期化する 文字列の先頭アドレスを返す PRINT内で、文字列を返す(数2(省略可)で長さ指定) 文字列の先頭アドレスを返す 「行の先頭に書くとラベルとなり、行番号の代わりとして使える(GOTO @LOOPなど)※前方一致 画面表示非表示を切り替えるので画面表示を停止し処理局態化(FBで表示)、省略可能な数2でVIDEOの時CPUクロックを1/数2に変更し省電力化、数1/52の倍数で自具を版、数/が32に上で拡大モード IchigoJamをリセットする プログラムを作止する (ボタンを押すと起動し、LRUN0を実行する) 数1: シリアル出力放定(のオフ、1: PRINTのみ、2: PRINTLO(CLS/SCROLL、3: PRINTのみの放行コードVin、+4で入力エコーバック、+8で画面表示 OFF、初期値2)、数2: シリアル受信該定(の:オフ、1: オン、+2: ESC無効、+4: CR変換(13 → 10) 省略時1) シリアル通信速度を変更する(ので初期値の115,2000ps、-1: 5/000ps、-2: 38400bps、-100以下指定で指定した数の-100倍bpsに指定、-2304(2300bps)、数2で120の通信速度放変度)、単位がLRV 4400kHz) OKやエラーメッセージの表示有無を切り替える(数、1:表示、2: 非表示、省略で1) I2Cで周辺機器から読み込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(数4)数5は省略可、コマンド送信が1byteの時数3を省略可) I2Cで周辺機器に書き込む 12Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ(数4)数5は省略可、コマンド送信が1byteの時数3を省略可) sakura.loモジュールへ数をチャンネルので即時送信する	?VER() RENUM LRUN 1 ?FILE() ?LINE() SRND 0 HELP ?PEEK(#700) POKE #700,#FF COPY #900,0,256 CLP A="ABC" PRINT STR\$(A) PRINT LEN("ABC") @LOOP VIDEO 0 RESET SLEEP UART 0 BPS 9600 OK 2 R=I2CR(#50,#114A,2,#114C,2) R=I0T.IN() IoT.OUT 100