TABLE 1A: GENERAL OPERATORS

Lvl	*Operator*	*Placed*	*Asc*	*Purpose*
0	`,`	between	left	separating values to work out
1	`=`	between	right	set equal to
5	`+`	between	left	16-bit signed addition
	`-`	between	left	16-bit signed subtraction
6	`*`	between	left	16-bit signed multiplication
	`/`	between	left	16-bit signed integer division
	`%`	between	left	16-bit signed remainder
	`&`	between	left	bitwise AND
	`|`	between	left	bitwise OR
	`~`	before		bitwise NOT
7	`->`	between	left	byte array entry
	`-->`	between	left	word array entry
8	`-`	before		16-bit signed negation
9	`++`	before		add 1 to then read
	`++`	after		read then add 1 to
	`--`	before		subtract 1 from then read
	`--`	after		read then subtract 1 from
10	`.&`	between	left	property array
	`.#`	between	left	property array size
11	`(...)`	after		routine call
12	`.`	between	left	property value
13	`::`	between	left	"superclass" operator

• "Lvl" refers to precedence level: thus `*`, on level 6, binds more tightly than `+`, down on level 5, so that `1+2*3` means `1+(2*3)`.

• `-` is "left associative", so `a-b-c` means `(a-b)-c`. `=` is "right associative", so `v1=v2=7` means `v1=(v2=7)`, setting both variables equal to 7.

• Although the table of operators has been divided over two pages, conditions and expressions can be freely mixed. When a condition is used as a value, it is always `true` (1) or `false` (0). When a value is used as a condition, any non-zero value is considered true, and only zero is considered false.

Table 1B: Condition Operators

Lvl	*Operator*	*Placed*	*Asc*	*Purpose*
2	`&&`	between	left	one condition AND another
	`||`	between	left	one condition OR another
	`~~`	before		this condition NOT true
3	`==`	between	*none*	equal to?
	`~=`	between	*none*	not equal to?
	`>`	between	*none*	greater than?
	`>=`	between	*none*	greater than or equal to?
	`<`	between	*none*	less than?
	`<=`	between	*none*	less than or equal to?
	`has`	between	*none*	object has this attribute?
	`hasnt`	between	*none*	object hasn't this attribute?
	`in`	between	*none*	first obj a child of second?
	`notin`	between	*none*	first obj not a child of second?
	`ofclass`	between	*none*	obj inherits from class?
	`provides`	between	*none*	obj provides this property?
4	`or`	between	left	separating alternative values

• Conditions have no associativity and if you type `a==b==c` then Inform will ask you to add brackets for clarity.

• In the condition `(C1 && C2)`, Inform decides on `C1` first: if `C1` is `false` then `C2` is never considered at all. Similarly, if `C1` is `true` then `(C1 || C2)` must be `true` and `C2` is never considered.

TABLE 2A: LOWER ZSCII CHARACTER SET

	+0	+1	+2	+3	+4	+5	+6	+7
0								
8	*del*	tab		em		new		
16								
24								*esc*
32	sp	!	"	#	$	%	&	’
40	(	)	⋆	+	,	-	.	/
48	0	1	2	3	4	5	6	7
56	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G
72	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W
88	X	Y	Z	[	\	]	^	_
96	‘	a	b	c	d	e	f	g
104	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	

• To convert a character to a ZSCII value, add the numbers in the same row and column. For instance, the Inform constant `'J'` is 72 plus 2 equals 74.

• Blank boxes indicate that no character exists with that value. The value will never be read from the keyboard and it is an error to try to `print (char)` it.

• Italicised entries can be read from the keyboard but not printed.

• “em” (an em-space) and “tab” (a tab-skip) are print-only, and only available if Inform is compiling a Version 6 game.

• ZSCII does not (normally) have “smart quotes”, that is, different characters for opening and closing quotations “ and ”. Some interpreters automatically smarten them when printed, though. And ZSCII does have ≪French≫ and ≫German≪ quotation marks (see table 2B).

TABLE 2B: HIGHER ZSCII CHARACTER SET

	+0	+1	+2	+3	+4	+5	+6	+7
128		↑	↓	←	→	*f1*	*f2*	*f3*
136	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*	*f10*	*f11*
144	*f12*	*k0*	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*
152	*k7*	*k8*	*k9*	ä `@:a`	ö `@:o`	ü `@:u`	Ä `@:A`	Ö `@:O`
160	Ü `@:U`	ß `@ss`	» `@>>`	« `@<<`	ë `@:e`	ï `@:i`	ÿ `@:y`	Ë `@:E`
168	Ï `@:I`	á `@’a`	é `@’e`	í `@’i`	ó `@’o`	ú `@’u`	ý `@’y`	Á `@’A`
176	É `@’E`	Í `@’I`	Ó `@’O`	Ú `@’U`	Ý `@’Y`	à `@‘a`	è `@‘e`	ì `@‘i`
184	ò `@‘o`	ù `@‘u`	À `@‘A`	È `@‘E`	Ì `@‘I`	Ò `@‘O`	Ù `@‘U`	â `@^a`
192	ê `@^e`	î `@^i`	ô `@^o`	û `@^u`	Â `@^A`	Ê `@^E`	Î `@^I`	Ô `@^O`
200	Û `@^U`	å `@oa`	Å `@oA`	ø `@\o`	Ø `@\O`	ã `@~a`	ñ `@~n`	õ `@~o`
208	Ã `@~A`	Ñ `@~N`	Õ `@~O`	æ `@ae`	Æ `@AE`	ç `@,c`	Ç `@,C`	th `@th`
216	eth `@et`	Th `@Th`	Eth `@Et`	£ `@LL`	œ `@oe`	Œ `@OE`	¡ `@!!`	¿ `@??`
224								
232								
240								
248					*men*	*dbl*	*clk*	

- The cursor keys, function keys, numeric keypad keys and mouse clicks (menu click, double click, single click) are read-only. Mouse support is available only to a Version 6 game.
- The given escape-character sequences can be typed into source code. For instance `print "@AEsop";` prints “Æsop”.
- “Eth” and “Th(orn)” are Icelandic characters.
- Characters 155 to 251 are configurable using the directive `Zcharacter`, which can in principle move any Unicode character in. See §36.

TABLE 3: COMMAND LINE SWITCHES

Sw	*To*	*Meaning*
`-d*`	0 to 2	contract double spaces: never (0), after full stops (1) after full stops, exclamation and question marks (2)
`-e`	off/on	economise by using the declared abbreviations
`-g*`	0 to 2	traces calls: none (0), all outside libraries (1), all (2)
`-i`	off/on	ignore default switches set within the file
`-k`	off/on	output Infix debugging information (and switch `-D` on)
`-r`	off/on	record all the text to a transcript file
`-v*`	3 to 8	compile to this Version of story file (default 5)
`-C*`	0 to 9	source is ASCII (0), or ISO 8859-1 to -9 (default 1)
`-D`	off/on	automatically include library's debugging features
`-F*`	0 or 1	use temporary files to reduce memory consumption
`-M`	off/on	compile as a Module for future linking
`-S`	on/off	compile strict error-checking at run-time (on by default)
`-U`	off/on	link in precompiled library modules
`-X`	off/on	include the Infix debugger
`-a`	off/on	trace assembly-language (without hex dumps; see `-t`)
`-c`	off/on	more concise error messages
`-f`	off/on	frequencies mode: show how useful abbreviations are
`-h*`	on/1/2	print help information: on filenaming (1), switches (2)
`-j`	off/on	list objects as constructed
`-l`	off/on	list every statement run through Inform
`-m`	off/on	say how much memory has been allocated
`-n`	off/on	print numbers of properties, attributes and actions
`-o`	off/on	print offset addresses
`-p`	off/on	give percentage breakdown of story file
`-q`	off/on	keep quiet about obsolete usages
`-s`	off/on	give statistics
`-t`	off/on	trace assembly-language (with full hex dumps; see `-a`)
`-u`	off/on	work out most useful abbreviations (very very slowly)
`-w`	off/on	disable warning messages
`-x`	off/on	print a hash # for every 100 lines compiled
`-y`	off/on	trace linking system
`-z`	off/on	print memory map of the Z-machine
`-E*`	0, 1, 2	errors in Acorn (0), Microsoft (1) or Mac (2) style

- The lower group has no effect except on what is printed out.
- The * stands for a decimal digit, `0` to `9`. You can also clear any switch with a tilde, so `-~x` turns `-x` off.

TABLE 4: STATEMENTS

```
box ⟨line-1⟩ ⟨line-2⟩ ... ⟨line-n⟩
break
continue
do ⟨code block⟩ until ⟨condition⟩
font on or off
for (⟨initial code⟩:⟨condition to carry on⟩:⟨update code⟩) ⟨code block⟩
give ⟨object⟩ ⟨attribute-1⟩ ... ⟨attribute-n⟩
if ⟨condition⟩ ⟨code block⟩
if ⟨condition⟩ ⟨code block⟩ else ⟨code-block⟩
jump ⟨label⟩
move ⟨object⟩ to ⟨destination⟩
new_line
objectloop ⟨condition choosing objects⟩ ⟨code block⟩
print ⟨list of printing specifications⟩
print_ret ⟨list of printing specifications⟩
remove ⟨object⟩
return ⟨optional value⟩
rfalse
rtrue
spaces ⟨number of spaces to print⟩
string ⟨number⟩ ⟨text⟩
style roman or bold or underline or reverse or fixed
switch (⟨value⟩) ⟨block of cases ... default: ...⟩
while ⟨condition⟩ ⟨code-block⟩
```

- Statements must be given in lower case.
- A statement beginning with a double-quoted string instead of a keyword like `if` is taken as a `print_ret` statement.
- Code blocks consist of either a single statement or a group of statements enclosed in braces `{` and `}`.
- The following low-level statements should not be used for Inform games:

```
inversion
quit
read ⟨text-buffer⟩ ⟨parsing-buffer⟩
restore ⟨label⟩
save ⟨label⟩
```

TABLE 5: DIRECTIVES

`Abbreviate` ⟨word-1⟩ . . . ⟨word-n⟩
`Array` ⟨new-name⟩ ⟨type⟩ ⟨initial values⟩
`Attribute` ⟨new-name⟩
`Class` ⟨new-name⟩ ⟨body of definition⟩
`Constant` ⟨new-name⟩ = ⟨value⟩
`Default` ⟨possibly-new-name⟩
`End`
`Endif`
`Extend` ⟨grammar extension⟩
`Global` ⟨new-name⟩ = ⟨value⟩
`Ifdef` ⟨symbol-name⟩
`Ifndef` ⟨symbol-name⟩
`Ifnot`
`Iftrue` ⟨condition⟩
`Iffalse` ⟨condition⟩
`Import` ⟨list of imported goods⟩
`Include` ⟨source code filename⟩
`Link` ⟨module filename⟩
`Lowstring` ⟨text⟩
`Message` ⟨message-type⟩ ⟨diagnostic-message⟩
`Object` ⟨header⟩ ⟨body of definition⟩
`Property` ⟨new-name⟩
`Release` ⟨number⟩
`Replace` ⟨routine-name⟩
`Serial` "⟨serial number⟩"
`Switches` ⟨list of switches⟩
`Statusline score` *or* `time`
`System_file`
`Verb` ⟨verb-definition⟩
`Zcharacter` etc.

• `Nearby` is an obsolete abbreviation for `Object ->`, now deprecated. A few other directives, `Dictionary`, `Fake_action`, `Ifv3`, `Ifv5`, `Stub`, `Trace` and `Version`, are either also obsolete or for compiler maintenance only.

TABLE 6A: ACTIONS PROVIDED BY THE LIBRARY: GROUP 1

Action	*Typically produced by*	*Notes*
`Pronouns`	"pronouns"	lists settings of "it" and so on
`Quit`	"quit"	
`Restart`	"restart"	
`Restore`	"restore"	
`Save`	"save"	
`Verify`	"verify"	checks story file integrity
`ScriptOn`	"script on"	
`ScriptOff`	"script off"	
`NotifyOn`	"notify on"	score change notification on
`NotifyOff`	"notify off"	and off
`Places`	"places"	list places `visited`
`Objects`	"objects"	list objects `moved`
`Score`	"score"	
`FullScore`	"fullscore"	full breakdown of score
`Version`	"version"	prints version numbers
`LMode1`	"brief"	normal room descriptions
`LMode2`	"verbose"	always full room descriptions
`LMode3`	"superbrief"	always abbreviated

• A number of other group 1 actions are present in a game compiled with the `-D` "Debugging" switch. These actions come and go with different library releases and their presence shouldn't be relied on. See the library's "Grammar" file to see the current set.

• The library also defines four fake actions which have nothing to do with the world model. `TheSame` and `PluralFound` are defined by the parser as ways for the program to communicate with it. `Miscellany` and `Prompt` are defined as slots for `LibraryMessages`.

Table 6B: Actions Provided By The Library: Group 2

Action	*Typically produced by*	*Notes*
`Look`	"look"	
`Examine`	"examine fish"	
`Search`	"look inside cup"	
`Inv`	"inventory"	
`InvTall`	"inventory tall"	*becomes* `Inv`
`InvWide`	"inventory wide"	*becomes* `Inv`
`Take`	"take fish"	*KS*
`Drop`	"drop fish"	*KS*
`Remove`	"take dice from cup"	*KS*
`PutOn`	"put cup on board"	*KS*
`Insert`	"put dice in cup"	*KS*
`LetGo`	*fake*	*caused by* `Remove`
`Receive`	*fake*	*caused by* `PutOn` *and* `Insert`
`Empty`	"empty sack"	*becomes* `EmptyT d_obj`
`EmptyT`	"empty bag on box"	*for each item inside, becomes* `Remove` *then* `Drop`/`PutOn`/`Insert`
`Transfer`	"transfer egg to box"	*becomes* `Drop`/`PutOn`/`Insert`
`Go`	"north"	*KS* *special rules apply: see* §15
`Enter`	"enter cage"	*KS* *can become* `Go` *if into a* `door`
`GetOff`	"get off table"	*KS*
`GoIn`	"enter"	*becomes* `Go in_obj`
`Exit`	"exit"	*KS* *can become* `Go out_obj`
`Unlock`	"unlock door"	*KS*
`Lock`	"lock door"	*KS*
`SwitchOn`	"switch radio on"	*KS*
`SwitchOff`	"switch radio off"	*KS*
`Open`	"open door"	*KS*
`Close`	"close door"	*KS*
`Disrobe`	"take hat off"	*KS*
`Wear`	"wear hat"	*KS*
`Eat`	"eat fish"	*KS*
`Wait`	"wait"	

• Actions marked *KS* run "silently" when the library's variable `keep_silent` is set `true`. This means that if successful they print nothing: if unsuccessful, however, they print text as normal.

• `Look` and `Examine` actions send `after` messages after printing descriptions. `Search` sends `after` when the search is known to be possible but before the result is printed.

Table 6C: Actions Provided By The Library: Group 3

Action	*Typically produced by*	*Notes*
`LookUnder`	"look under doormat"	
`Listen`	"listen [to tape]"	`noun` *can be* `nothing`
`Taste`	"taste marinade"	
`Touch`	"touch paint"	
`Pull`	"pull trolley"	
`Push`	"push trolley"	
`Wave`	"wave wand"	
`Turn`	"turn dial"	
`PushDir`	"push trolley north"	*special rules apply: see §15*
`ThrowAt`	"throw dart at board"	
`ThrownAt`	*fake*	*caused by* `ThrowAt`
`JumpOver`	"jump over fence"	
`Tie`	"tie rope [to hook]"	`second` *can be* `nothing`
`Drink`	"drink absinthe"	
`Fill`	"fill bottle"	
`Attack`	"fight soldiers"	
`Swing`	"swing on rope"	
`Blow`	"blow pipe"	
`Rub`	"clean table"	
`Set`	"set trap"	
`SetTo`	"set timer to 10"	`second` *not an object*
`Buy`	"buy ice cream"	
`Climb`	"climb ladder"	
`Squeeze`	"squash tomato"	
`Burn`	"burn papers [with match]"	`second` *can be* `nothing`
`Dig`	"dig lawn [with spade]"	`second` *can be* `nothing`
`Cut`	"cut paper"	
`Consult`	"look up fish in book"	sets `noun` and the topic
`Tell`	"tell jemima about austin"	sets `noun` and the topic
`Answer`	"say confirmed to avon"	sets `noun` and the topic
`Ask`	"ask jemima about isaac"	sets `noun` and the topic
`Give`	"give coin to troll"	
`Show`	"show pass to guard"	
`AskFor`	"ask jemima for daisies"	
`WakeOther`	"wake sleeper"	
`Kiss`	"kiss jemima"	

Table 6C (Continued)

Action	*Typically produced by*	*Notes*
`Sleep`	"sleep"	
`Sing`	"sing"	
`WaveHands`	"wave"	*see also* `Wave`
`Swim`	"swim", "dive"	
`Sorry`	"sorry"	
`Strong`	*very rude words*	
`Mild`	*fairly rude words*	
`Jump`	"jump"	*see also* `JumpOver`
`Think`	"think"	
`Smell`	"smell coffee"	`noun` *can be* `nothing`
`Pray`	"pray"	
`VagueGo`	"go"	
`Yes`	"yes"	
`No`	"no"	
`Wake`	"wake up"	*see also* `WakeOther`

Table 6D: Actions Sent To Life Rules

Action	*Typically produced by*
`Answer`	"say yes to cashier"
`Ask`	"ask woman about plutonium"
`Attack`	"fight soldiers"
`Give`	"give coin to charon"
`Kiss`	"kiss jemima"
`Order`	"thorin, go west"
`Show`	"show pass to benton"
`Tell`	"tell paris about helen"
`ThrowAt`	"throw axe at dwarf"
`WakeOther`	"wake beauty up"