

1 IO

```
int printf (const char *format, ...); (stdio.h)
int scanf (const char *format, ...); (stdio.h)
int putchar(int c); (stdio.h)
int getchar(void); (stdio.h)
```

1.1 Escape sequences

Sequence	ASCII	Description
<code>\n</code>	10	new line
<code>\r</code>	13	carriage return
<code>\t</code>	09	horizontal tab
<code>\v</code>	11	vertical tab
<code>\f</code>	12	form feed (Cursor to start of next page)
<code>\b</code>	08	backspace
<code>\a</code>	07	bell
<code>\'</code>	39	apostrophe
<code>\"</code>	34	quote
<code>\\</code>	92	backslash
<code>\nnn</code>		char value in octal (n = 0..7)
<code>\xhh</code>		char value in hex

1.2 Format

Sequence	Output
<code>%d</code> or <code>%i</code>	int
<code>%c</code>	character
<code>%e</code> or <code>%E</code>	double in format [-] d.ddd e±dd
<code>%f</code>	double in format [-] ddd.ddd
<code>%ot</code>	int as octal
<code>%s</code>	string
<code>%p</code>	As pointer address
<code>%u</code>	unsigned int
<code>%x</code> or <code>%X</code>	int as hex
<code>%%</code>	%

`%ni` output as int / flush right / n characters wide

`%0ni` output as int / flush right / n characters wide filled with 0

`%.nf` output as float / n digits after comma

`%+i` output as int / forces plus or minus sign

`%#x` int as hex / forces 0x before number

1.3 Clear stdin buffer

```
while(getchar() != '\n');
```

2 Procedure structure

- Pseudocode
- Flowchart
- structure chart

3 if...else...

- {...} is considered a code block
- <condition> ? <true block> : <false block>

4 switch...

- Mark when multiple cases point to one block with `//fall through` comment.

5 Syntax description

- EBNF (Erweiterte Backus-Naur-Form)
- Syntax diagram

6 For loop

```
{ init-statement ; while ( condition ) { statement expression ; } }
```

7 Priorities

Priority	Symbol	Associativity	Meaning
15	(Postfix) ++, --	L - R	Increment, Decrement
	(Postfix) --		Decrement
	()		Function call
	[]		Indexing
14	(Prefix) ++, --	R - L	Increment, Decrement
	+, - (positiv/negativ number)		positiv/negativ number
	!		logic not
	(Type)		type conversion
	sizeof		
13	*, /, %	L - R	Rechenoperation
12	+, -	L - R	Rechenoperation
10	<	L - R	kleiner
	<=		kleiner gleich
	>		grösser
	>=		grösser gleich
9	==	L - R	gleich
	!=		ungleich
5	&&	L - R	logisches UND
4		L - R	logisches ODER
3	?:	R - L	Bedingung
2	=	R - L	Zuweisung
	*=, /=, %=, +=, -=,		Kombinierte Zuweisung
1	,	L - R	Komma-Operator

7.1 Weird operators

`i++` → Value is returned before and then incremented

`++i` → Value is first incremented and then returned