1 grammer

- stmt: stmt;
- stmt: { stmt; ... }
- end;
- let expr;
- print expr;
- read expr;
- data raw_input;
- goto label;
- label:
- if (expr) stmt;
- if (expr) stmt; else stmt;
- while (expr) stmt;
- for(expr; expr; expr) stmt;
- do { stmt; } while (expr);

2 if command

1. if (—) [.1] —; [.2][.3]
2. if (—) {[.1] –run parser again }[.2][.3]
3. if (—) [.1]—; else { [.2] —; }[.3]
4. if (—) {[.1] –run parser again } else [.2] —; [.3]
5. if (—) {[.1] –run parser again } else {[.2] –run parser again }[.3]

3 while command

.1 while (—) [.2] —; [.3] [.4]

.1 while (—) { [.2] –run parser again } [.3][.4]

.1 while (—) [.2]—; else { [.3] —; }[.4]

.1 while (—) {[.2] –run parser again } else [.3] —; [.4]

.1 while (—) {[.2] –run parser again } else {[.3] –run parser again } [.4]

3.1 while representation

```
[.n1]  if !(−−−) goto  [.n3]
[.n2]  while body
[.f1]  if !(−−−) goto  [.n4]
[.g2]  goto  [.n2]
else    [.n3] { else body    };
[.n4]
```

[SOLVED] how to find) after if.