

This document shows only the grammar for the Y programming language for each part of the project. This information is contained in the project description, but some students find it handy to be able to print out just the grammar and refer to it as they work on each part.

INITIAL GRAMMAR

```
Program --> 'program' NAME ':' Block 'end'
Block --> {Statement}
Statement --> PrintStatement
PrintStatement --> 'print' Expression
Expression --> AddExpr
AddExpr --> Atom [('+' | '-') AddExpr]
Atom --> IntegerLiteral | '(' Expr ')'
```

PART 1

```
Program --> 'program' NAME ':' Block 'end'
Block --> {Statement}
Statement --> PrintStatement
PrintStatement --> 'print' Expression
Expression --> AddExpr
AddExpr --> MultExpr [('+' | '-') AddExpr]
MultExpr --> Atom [('*' | '/' | '%') MultExpr]
Atom --> IntegerLiteral | '(' Expr ')'
```

PART 2

```
Program --> 'program' NAME ':' Block 'end'
Block --> {Statement}
Statement --> PrintStatement | AssignStatement
PrintStatement --> 'print' Expression
AssignStatement --> NAME ':' Expression
Expression --> AddExpr | StringExpr
StringExpr --> StringLiteral
AddExpr --> MultExpr ['+' | '-'] AddExpr
MultExpr --> NegExpr ['*' | '/' | '%'] MultExpr
NegExpr --> '-' NegExpr | Atom
Atom --> IntegerLiteral | '(' Expression ')' | NAME
```

PART 3

```
Program --> 'program' NAME ':' Block 'end'
Block --> {Statement}
Statement --> PrintStatement | AssignStatement | WhileLoop
WhileLoop --> 'while' CondExpr ':' Block 'end'
PrintStatement --> 'print' Expression
AssignStatement --> Name ':' Expression
Expression --> CondExpr | StringExpr
StringExpr --> StringLiteral
CondExpr --> AddExpr ['=' | '<' | '<=' | '>' | '>=' | '<>'] AddExpr
AddExpr --> MultExpr ['+' | '-'] AddExpr
MultExpr --> NegExpr ['*' | '/' | '%'] MultExpr
NegExpr --> '-' NegExpr | Atom
Atom --> IntegerLiteral | '(' Expression ')' | Name
```