



Compilers Phase 1 Document

C-like language

Ahmed Sayed	sec 1 bn 12
Ahmed Mahmoud	sec 1 bn 6
Mohamed Nabawe	sec 2 bn 16
Mohamed Khaled	sec 2 bn 13



Lexer :

Keywords:

if, else, do, while, for, break, continue, switch, case, default, char, string, int, float ,
, bool, const, void, return, enum, print

Operators:

"==", "!=", ">=", "<=", "&&", "||"

Constants:

[-+*/%<>!{}&(),;=] , true, false, [0-9]+ , [0-9]*\.[0-9]+ ,
['][a-zA-Z](['], \"^[^']*\")*

Identifier:

[a-zA-Z_][0-9a-zA-Z_]*

Parser :

All the productions are in 'parser.y':

• Variables and Constants declaration:

C-like

```
int x =3; int z;
```

```
const float y=3.4;
```

And all other types will follow this structure

• Mathematical and logical expressions.

```
x= x+1 ;
```

```
y= x* 3 +2;
```

No power operator (^) or ++ or —

C does not have power operator 😊

- Assignment statement.

C-like

`x = y = z;` x and y will have the value of z

- If-then-else statement

Like C

```
if (x==3){}  
else if (x==4){}  
else {}  
x=4;
```

OR

```
if (x==3){}  
else {}  
x=4;
```

OR

```
if (x==3){}  
x=4;
```

OR

```
if (x==3)  
    y=5; // if body  
x=5;
```

- while loops

Like C

- repeat-until loops

Like C

- For loops

Like C

- switch statement.

Simplified a bit than C

Have to put curly braces for each case, for example:

```
switch (temp) :  
{      // switch statement can have an empty body  
    case 'a': {}  
    default : {} // not necessary to write default  
}
```

- Block structure (nested scopes where variables may be declared at the beginning of blocks).

Like C

- Enums

Like C

```
enum week {Saturday=0, Sunday, Tuesday=1};
```

- Functions

Like C but for the given data types only.

