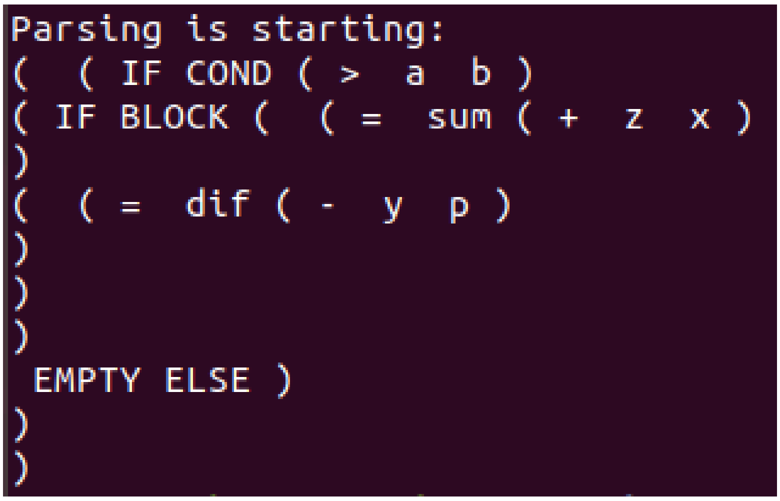
Project – Part 2 (Parser)

Execution examples:

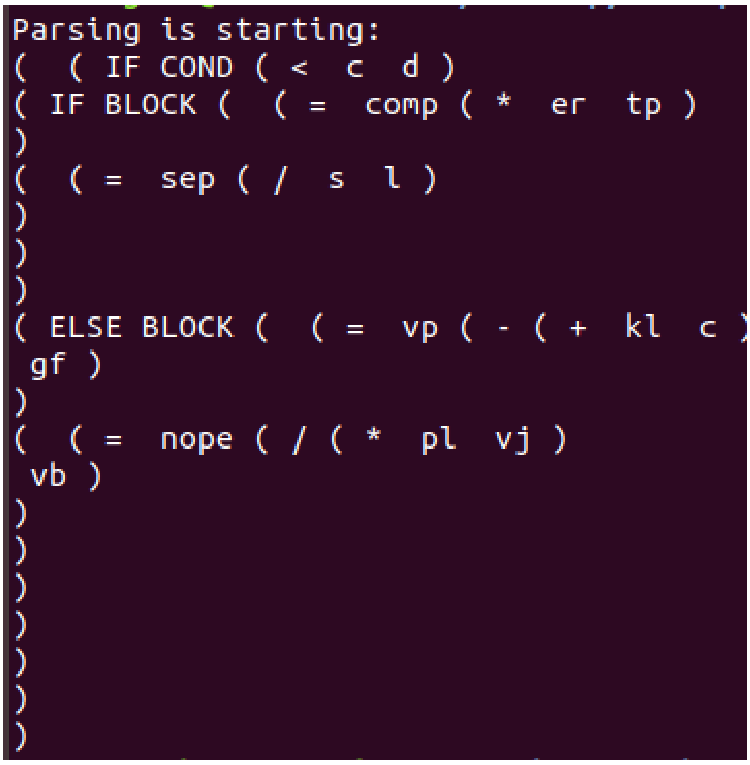
```
/* Loop: for, while, do while; if else */
```

```
/* Example #1 */

if(a > b) {

     sum = z + x;
     dif = y - p;
}
```



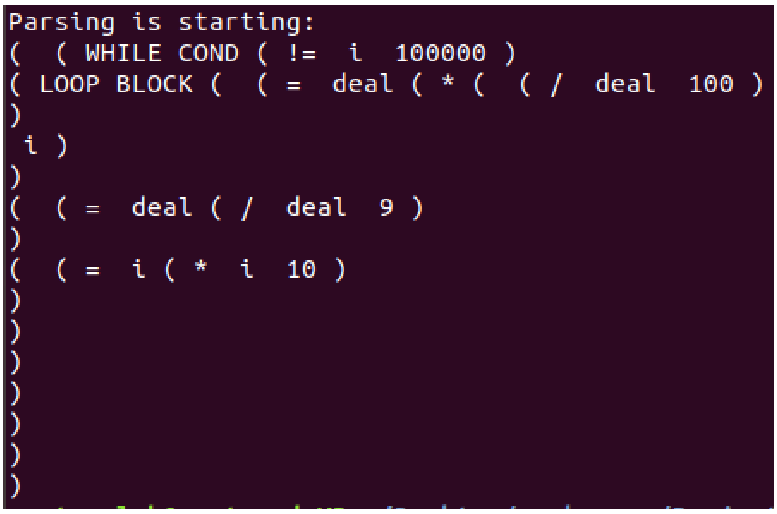
```
Parsing is starting:
(   ( IF COND ( >  a  b )
( IF BLOCK (  ( =  sum ( +  z  x )
)
(   ( =  dif ( -  y  p )
)
)
)
 EMPTY ELSE )
)
)
```

```
/* Example #2 */


if(c < d){

   comp = er * tp;
   sep = s / l;
}
else{

   vp = kl + c - gf;
   nope = pl * vj / vb;
}
```



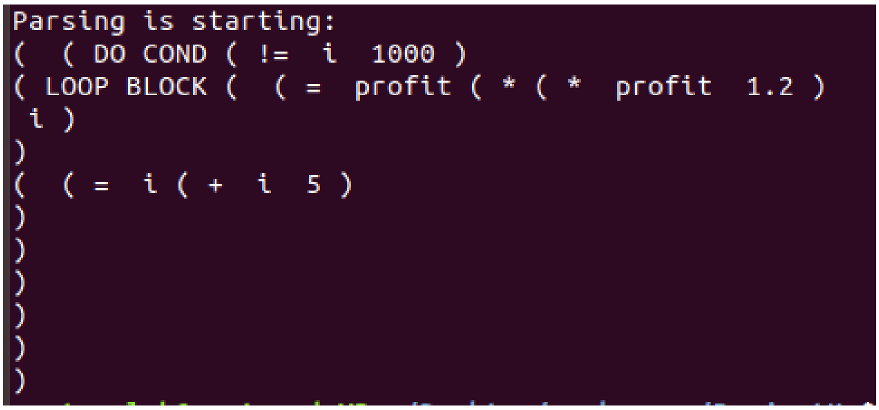
```
Parsing is starting:
(  ( IF COND ( <  c  d )
( IF BLOCK (  ( =  comp ( *  er  tp )
)
(  ( =  sep ( /  s  l )
)
)
)
( ELSE BLOCK (  ( =  vp ( - ( +  kl  c )
 gf )
)
(  ( =  nope ( / ( *  pl  vj )
 vb )
)
)
)
)
)
)
)
```

```
/* Example #3 */

while (i != 100000){

 deal = (deal % 100) * i;
        deal = deal / 9;
        i = i * 10;
}
```



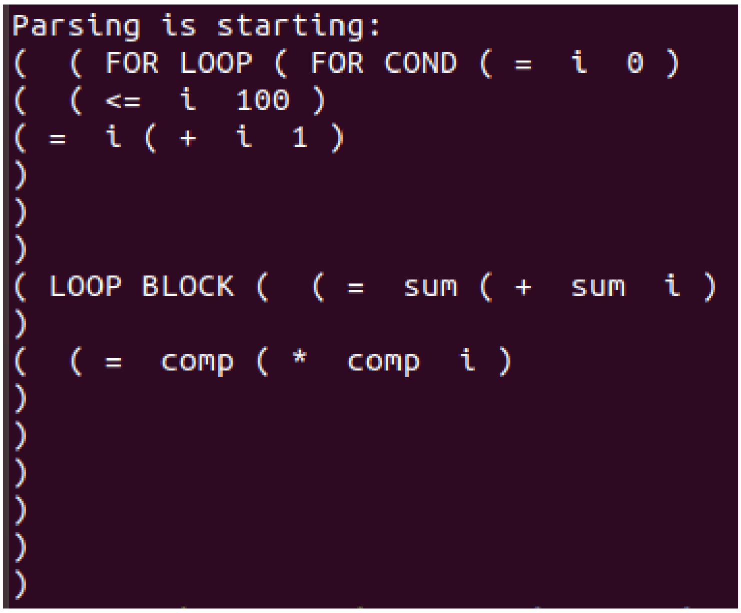
```
Parsing is starting:
(   ( WHILE COND ( !=  i  100000 )
( LOOP BLOCK (   ( =  deal ( * (  ( /  deal  100 )
)
 i )
)
(   ( =  deal ( /  deal  9 )
)
(   ( =  i ( *  i  10 )
)
)
)
)
)
)
)
```

```
/* Example #4 */

do{

profit = profit * 1.2 * i;
i = i + 5;
}
while (i != 1000)
```



```
Parsing is starting:
(   ( DO COND ( !=  i  1000 )
( LOOP BLOCK (   ( =  profit ( * ( *  profit  1.2 )
 i )
)
(   ( =  i ( +  i  5 )
)
)
)
)
)
)
```

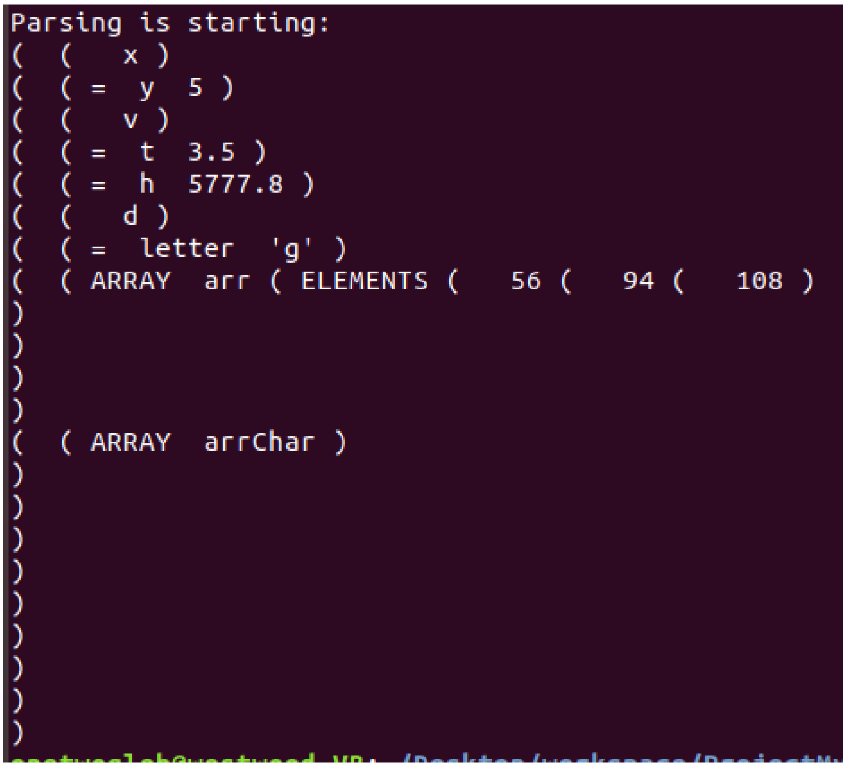
```
/* Example #5 */

for (int i = 0; i <= 100; i = i + 1 ){

          sum = sum + i;

          comp = comp * i;

}
```



```
Parsing is starting:
(  ( FOR LOOP ( FOR COND ( =  i  0 )
(  ( <=  i  100 )
( =  i ( +  i  1 )
)
)
)
( LOOP BLOCK (  ( =  sum ( +  sum  i )
)
(  ( =  comp ( *  comp  i )
)
)
)
)
)
)
```

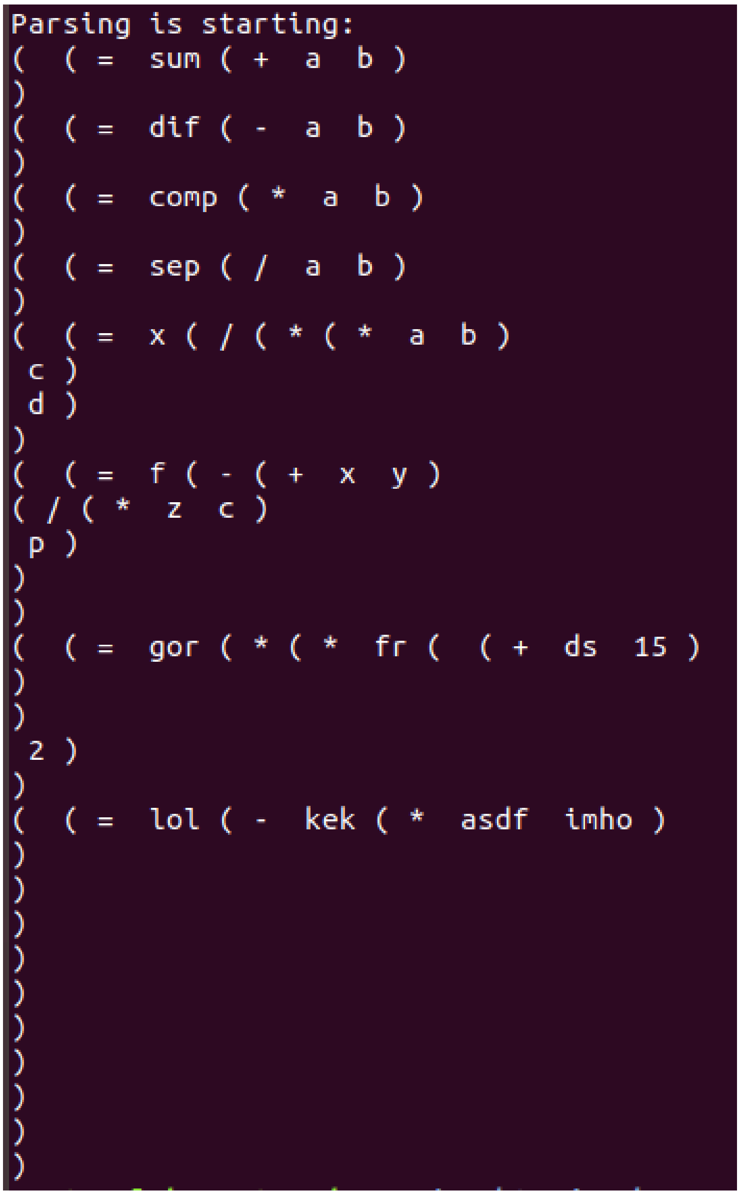
/* Different Types */

```
int x;
int y = 5;
float v;
float t = 3.5;
float h = 5777.8;
char d;
char letter = 'g';
int arr[] = {56, 94, 108};
char arrChar[];
```



```
Parsing is starting:
(   (    x )
(   ( =  y  5 )
(   (    v )
(   ( =  t  3.5 )
(   ( =  h  5777.8 )
(   (    d )
(   ( =  letter  'g' )
(   ( ARRAY  arr ( ELEMENTS (   56 (   94 (   108 )
)
)
)
)
(   ( ARRAY  arrChar )
)
)
)
)
)
)
)
)
)
```

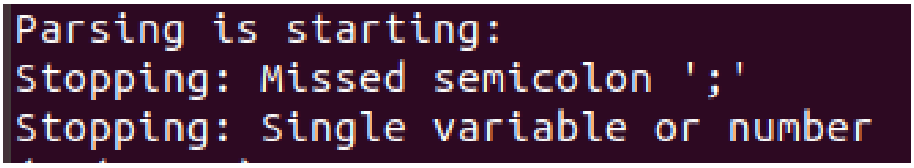
```
/* Math Operations */

sum = a + b;
dif = a - b;
comp = a * b;
sep = a / b;


int x = a * b * c / d;
f = x + y - z * c / p;
gor = fr * (ds + 15) * 2;
lol = kek - asdf * imho;
```



```
Parsing is starting:
(  ( =  sum ( +  a  b )
)
(  ( =  dif ( -  a  b )
)
(  ( =  comp ( *  a  b )
)
(  ( =  sep ( /  a  b )
)
(  ( =  x ( / ( * ( *  a  b )
 c )
 d )
)
(  ( =  f ( - ( +  x  y )
( / ( *  z  c )
 p )
)
)
(  ( =  gor ( * ( *  fr (  ( +  ds  15 )
)
)
 2 )
)
(  ( =  lol ( -  kek ( *  asdf  imho )
)
)
)
)
)
)
)
)
)
)
```

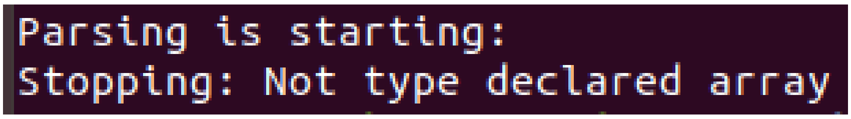
/* Error Checking */

```
/* Example #1 */
int x x;
y;
```



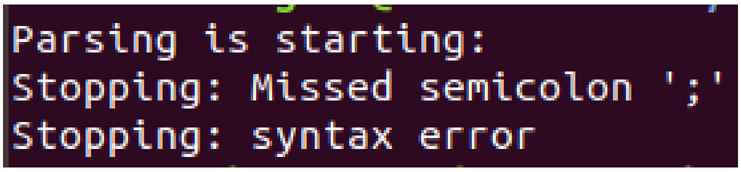
```
Parsing is starting:
Stopping: Missed semicolon ';'
Stopping: Single variable or number
```

```
/* Example #2 */

arr[] = {1, 2, 3};
```



```
Parsing is starting:
Stopping: Not type declared array
```

```
/* Example #3 */

y = 5 + 8
if(i>5
    {g = g+1; i = i+1;}
```



```
Parsing is starting:
Stopping: Missed semicolon ';'
Stopping: syntax error
```