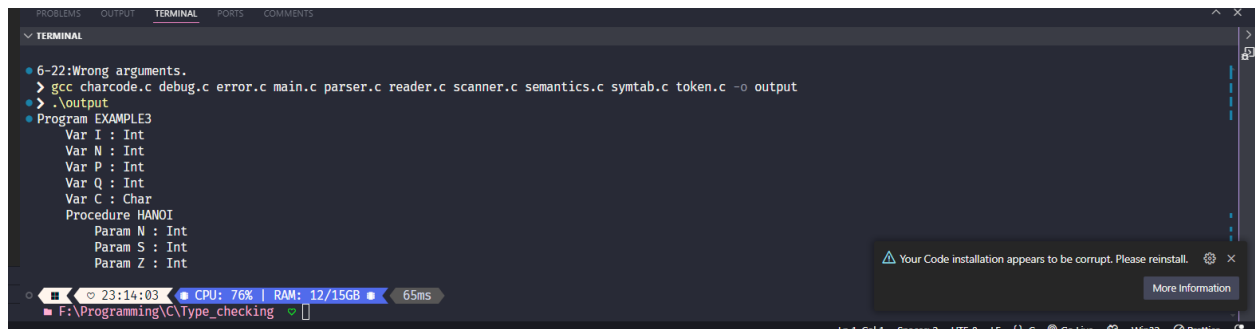


# TYPE CHECKING REPORT

Full Name: Dinh Viet Quang

Student ID: 20215235

## 1. Results (a part of the screen) received when executing project with example3.kpl.

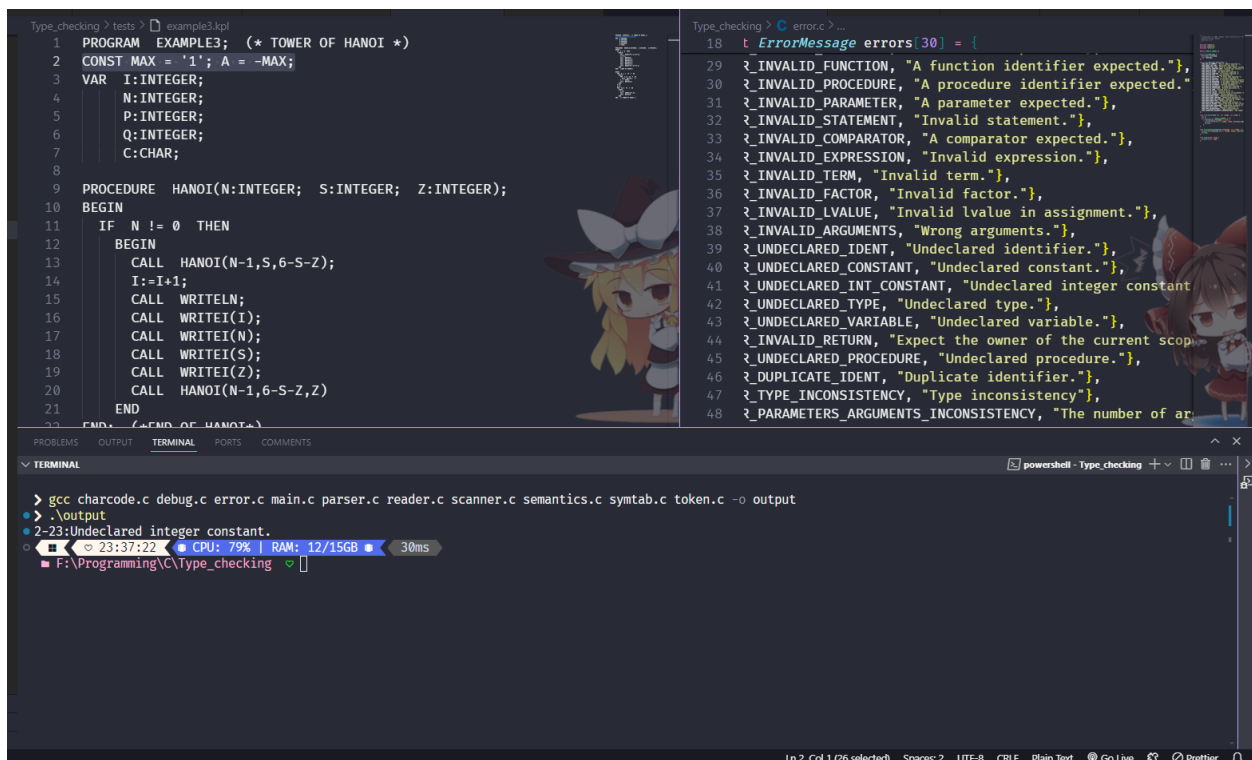


```
6-22:Wrong arguments.
> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c syntab.c token.c -o output
> .\output
Program EXAMPLE3
  Var I : Int
  Var N : Int
  Var P : Int
  Var Q : Int
  Var C : Char
  Procedure HANOI
    Param N : Int
    Param S : Int
    Param Z : Int
```

## 2. All kinds of errors that can be found by the project

### 2.1. ERR\_UNDECLARED\_INT\_CONSTANT

Error: CONST MAX = '1'; A = -MAX;;



```
1 PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
2 CONST MAX = '1'; A = -MAX;
3 VAR I:INTEGER;
4     N:INTEGER;
5     P:INTEGER;
6     Q:INTEGER;
7     C:CHAR;
8
9 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
10 BEGIN
11   IF N != 0 THEN
12     BEGIN
13       CALL HANOI(N-1,S,6-S-Z);
14       I:=I+1;
15       CALL WRITELN;
16       CALL WRITEI(I);
17       CALL WRITEI(N);
18       CALL WRITEI(S);
19       CALL WRITEI(Z);
20       CALL HANOI(N-1,6-S-Z,Z)
21     END
22   END;
23 END; (*END OF HANOI*)
```

```
18 t.ErrorMessage errors[30] = {
29   INVALID_FUNCTION, "A function identifier expected."},
30   INVALID_PROCEDURE, "A procedure identifier expected."},
31   INVALID_PARAMETER, "A parameter expected."},
32   INVALID_STATEMENT, "Invalid statement."},
33   INVALID_COMPARATOR, "A comparator expected."},
34   INVALID_EXPRESSION, "Invalid expression."},
35   INVALID_TERM, "Invalid term."},
36   INVALID_FACTOR, "Invalid factor."},
37   INVALID_LVALUE, "Invalid lvalue in assignment."},
38   INVALID_ARGUMENTS, "Wrong arguments."},
39   UNDECLARED_IDENT, "Undeclared identifier."},
40   UNDECLARED_CONSTANT, "Undeclared constant."},
41   UNDECLARED_INT_CONSTANT, "Undeclared integer constant",
42   UNDECLARED_TYPE, "Undeclared type."},
43   UNDECLARED_VARIABLE, "Undeclared variable."},
44   INVALID_RETURN, "Expect the owner of the current scope",
45   UNDECLARED_PROCEDURE, "Undeclared procedure."},
46   DUPLICATE_IDENT, "Duplicate identifier."},
47   TYPE_INCONSISTENCY, "Type inconsistency"},
48   PARAMETERS_ARGUMENTS_INCONSISTENCY, "The number of ar
```

### 2.2. ERR\_TYPE\_INCONSISTENCY

Error: Q:=C; (explain: Q is INTEGER, C is CHAR)

```

9 BEGIN
10 IF N != 0 THEN
11 BEGIN
12 END; (*END OF HANOI*)
13
14 BEGIN
15 FOR N := 1 TO 4 DO
16 BEGIN
17 FOR I:=1 TO 4 DO
18 CALL WRITEC(' ');
19 C := READC;
20 CALL WRITEC(C)
21 END;
22 P:=1;
23 Q:=C;
24 FOR N:=2 TO 4 DO
25 BEGIN
26 I:=0;
27 CALL HANOI(N,P,Q);
28 CALL WRITELN
29 END;
30 END;

```

```

18 t ErrorMessage errors[30] = {
39 <INVALID_ARGUMENTS>, wrong arguments. },
39 <UNDECLARED_IDENTIFIER>, "Undeclared identifier."},
40 <UNDECLARED_CONSTANT>, "Undeclared constant."},
41 <UNDECLARED_INT_CONSTANT>, "Undeclared integer constant",
42 <UNDECLARED_TYPE>, "Undeclared type."},
43 <UNDECLARED_VARIABLE>, "Undeclared variable."},
44 <INVALID_RETURN>, "Expect the owner of the current scope",
45 <UNDECLARED_PROCEDURE>, "Undeclared procedure."},
46 <DUPLICATE_IDENT>, "Duplicate identifier."},
47 <TYPE_INCONSISTENCY>, "Type inconsistency"},
48 <PARAMETERS_ARGUMENTS_INCONSISTENCY>, "The number of arguments",
49
50
51 error(ErrorMessage err, int lineNo, int colNo) {
52 i;
53 (i = 0 ; i < NUM_OF_ERRORS; i++)
54 f (errors[i].errorCode == err) {
55 printf("%d-%d:%s\n", lineNo, colNo, errors[i].message)
56 exit(0);
57
58

```

```

> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
32-6:Type inconsistency
CPU: 78% | RAM: 12/15GB | 30ms

```

Error: A(I.) := 'a' + 1; (A is an integer array, however, in line 16, each element of A is assigned with a character value)

```

1 PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
2 TYPE B = INTEGER;
3 VAR
4 I:INTEGER;
5 N:INTEGER;
6 P:INTEGER;
7 Q:INTEGER;
8 C:CHAR;
9 A: ARRAY(1..10) OF B;
10
11 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
12 BEGIN
13 IF N != 0 THEN
14 BEGIN
15 CALL HANOI(N-1,S,6-S-Z);
16 I:=I+1;
17 A(I.) := 'a' + 1;
18 CALL WRITELN;
19 CALL WRITEI(I);
20 CALL WRITEI(N);
21 CALL WRITEI(S);
22 CALL WRITEI(Z);
23 CALL HANOI(N-1,6-S-Z,Z);
24 END;
25 END;

```

```

18 t ErrorMessage errors[30] = {
39 <INVALID_ARGUMENTS>, wrong arguments. },
39 <UNDECLARED_IDENTIFIER>, "Undeclared identifier."},
40 <UNDECLARED_CONSTANT>, "Undeclared constant."},
41 <UNDECLARED_INT_CONSTANT>, "Undeclared integer constant",
42 <UNDECLARED_TYPE>, "Undeclared type."},
43 <UNDECLARED_VARIABLE>, "Undeclared variable."},
44 <INVALID_RETURN>, "Expect the owner of the current scope",
45 <UNDECLARED_PROCEDURE>, "Undeclared procedure."},
46 <DUPLICATE_IDENT>, "Duplicate identifier."},
47 <TYPE_INCONSISTENCY>, "Type inconsistency"},
48 <PARAMETERS_ARGUMENTS_INCONSISTENCY>, "The number of arguments",
49
50
51 error(ErrorMessage err, int lineNo, int colNo) {
52 i;
53 (i = 0 ; i < NUM_OF_ERRORS; i++)
54 f (errors[i].errorCode == err) {
55 printf("%d-%d:%s\n", lineNo, colNo, errors[i].message)
56 exit(0);
57
58

```

```

> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
16-23:Type inconsistency

```

Error: I:=I+'a';

```

8 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
9 BEGIN
10 IF N != 0 THEN
11 BEGIN
12 CALL HANOI(N-1,S,6-S-Z);
13 I:=I+'a';
14 CALL WRITELN;
15 CALL WRITEI(I);
16 CALL WRITEI(N);
17 CALL WRITEI(S);
18 CALL WRITEI(Z);
19 CALL HANOI(N-1,6-S-Z,Z);
20 END;
21 END; (*END OF HANOI*)

```

```

44 <INVALID_RETURN>, "Expect the owner of the current scope",
45 <UNDECLARED_PROCEDURE>, "Undeclared procedure."},
46 <DUPLICATE_IDENT>, "Duplicate identifier."},
47 <TYPE_INCONSISTENCY>, "Type inconsistency"},
48 <PARAMETERS_ARGUMENTS_INCONSISTENCY>, "The number of arguments",
49
50
51 error(ErrorMessage err, int lineNo, int colNo) {
52 i;
53 (i = 0 ; i < NUM_OF_ERRORS; i++)
54 f (errors[i].errorCode == err) {
55 printf("%d-%d:%s\n", lineNo, colNo, errors[i].message)
56 exit(0);
57
58

```

```

> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
13-13:Type inconsistency
CPU: 85% | RAM: 13/15GB | 29ms

```

Error: FOR N:=2 TO 'z' DO

```
9 BEGIN
10 IF N != 0 THEN
20 END
21 END; (*END OF HANOI*)
22
23 BEGIN
24 FOR N := 1 TO 4 DO
25 BEGIN
26 FOR I:=1 TO 4 DO
27 CALL WRITEC(' ');
28 C := READC;
29 CALL WRITEC(C)
30 END;
31 P:=1;
32 Q:=2;
33 FOR N:=2 TO 'z' DO
34 BEGIN
35 I:=0;
36 CALL HANOI(N,P,Q);
37 CALL WRITELN
38 END
END
```

```
18 t ErrorMessage errors[30] = {
38 <_INVALID_ARGUMENTS, wrong arguments. },
39 <_UNDECLARED_IDENT, "Undeclared identifier."},
40 <_UNDECLARED_CONSTANT, "Undeclared constant."},
41 <_UNDECLARED_INT_CONSTANT, "Undeclared integer constant"},
42 <_UNDECLARED_TYPE, "Undeclared type."},
43 <_UNDECLARED_VARIABLE, "Undeclared variable."},
44 <_INVALID_RETURN, "Expect the owner of the current scope"},
45 <_UNDECLARED_PROCEDURE, "Undeclared procedure."},
46 <_DUPLICATE_IDENT, "Duplicate identifier."},
47 <_TYPE_INCONSISTENCY, "Type inconsistency"},
48 <_PARAMETERS_ARGUMENTS_INCONSISTENCY, "The number of arguments and the number of parameters are inconsistent"};
49
50
51 error(ErrorMessage err, int lineNo, int colNo) {
52 i;
53 (i = 0; i < NUM_OF_ERRORS; i++)
54 if (errors[i].errorCode == err) {
55 printf("%d-%d:%s\n", lineNo, colNo, errors[i].message);
56 exit(0);
57
58
```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS

TERMINAL

```
> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
33-19:Type inconsistency
01:06:57 CPU: 84% RAM: 13/15GB 29ms
F:\Programming\C\Type_checking
```

## 2.3. ERR\_PARAMETERS\_ARGUMENTS\_INCONSISTENCY

Error: CALL HANOI(N-1); (explain: HANOI function needs 3 arguments as input, but in this case, the recursively called function contains only 1 argument)

```
1 PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
2 VAR I:INTEGER;
3 N:INTEGER;
4 P:INTEGER;
5 Q:INTEGER;
6 C:CHAR;
7
8 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
9 BEGIN
10 IF N != 0 THEN
11 BEGIN
12 CALL HANOI(N-1,S,6-S-Z);
13 I:=I+1;
14 CALL WRITELN;
15 CALL WRITEI(I);
16 CALL WRITEI(N);
17 CALL WRITEI(S);
18 CALL WRITEI(Z);
19 CALL HANOI(N-1)
20 END
21 END; (*END OF HANOI*)
```

```
18 t ErrorMessage errors[30] = {
38 <_INVALID_ARGUMENTS, wrong arguments. },
39 <_UNDECLARED_IDENT, "Undeclared identifier."},
40 <_UNDECLARED_CONSTANT, "Undeclared constant."},
41 <_UNDECLARED_INT_CONSTANT, "Undeclared integer constant"},
42 <_UNDECLARED_TYPE, "Undeclared type."},
43 <_UNDECLARED_VARIABLE, "Undeclared variable."},
44 <_INVALID_RETURN, "Expect the owner of the current scope"},
45 <_UNDECLARED_PROCEDURE, "Undeclared procedure."},
46 <_DUPLICATE_IDENT, "Duplicate identifier."},
47 <_TYPE_INCONSISTENCY, "Type inconsistency"},
48 <_PARAMETERS_ARGUMENTS_INCONSISTENCY, "The number of arguments and the number of parameters are inconsistent"};
49
50
51 error(ErrorMessage err, int lineNo, int colNo) {
52 i;
53 (i = 0; i < NUM_OF_ERRORS; i++)
54 if (errors[i].errorCode == err) {
55 printf("%d-%d:%s\n", lineNo, colNo, errors[i].message);
56 exit(0);
57
58
```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS

TERMINAL

```
> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
2-23:Undeclared integer constant.
> gcc charcode.c debug.c error.c main.c parser.c reader.c scanner.c semantics.c sytab.c token.c -o output
> .\output
20-0:The number of arguments and the number of parameters are inconsistent.
23:39:52 CPU: 81% RAM: 12/15GB 57ms
F:\Programming\C\Type_checking
```