

Computer Science and Engineering
IIIT Kalyani, West Bengal

Compilers Design Laboratory (Spring: 2018 - 2019)

3rd Year CSE: 6th Semester

Assignment - 7 Marks: 10
Assignment Out: 25th October, 2019 Report on or before: 1st November, 2019

Consider the following fragment of the grammar of **assignment 6**.

```
P → DL SL
DL → DL D | ε
D → VL : TY
TY → int | real
VL → VL id | id
SL → S SL | S
S → AS | IOS | nop
AS → id = E
PE → E | str
E → E + E | E - E | E * E | E / E | - E | ( E ) | id | ic | fc
```

Within the *bison* specification of the grammar you need to write C++ code fragments to take *semantic action* and *3-address code generation* (or *interpretation*). This will be done in **phases**. In the **first phase** a *symbol table* is created and variables are inserted in it during its deceleration. This also helps to raise error signal for re-declaration of same variable and use of a variable without declaration.

1. *Symbol table*: You need to implement a symbol table. Make it as simple possible. Each entry of it stores an identifier, its type, size, displacement on stack frame etc.¹. Implement it as a data type in separate files `symTab.c++` and `symTab.h`. Essential functions are initialization (class constructor if you use class), search, insert and print-table. a suggested type is as follows. But you may have your own choice.

```
typedef struct {
    char id[ID_LEN];
    char type;
    int size;
    int displacement;
} symTabEntry;

class symTab {
    symTabEntry ste[ST_SIZE];
    int nextIndex;
    int displacement;

public:
    symTab();
    // Other member fubctions
};
```

2. A declared variable is inserted in the table along with its type and other information.
3. If a variable is declared for the second time, an error message is printed.
4. Other than the variable deceleration write semantic action (C++ code) corresponding to the rule $E \rightarrow id$. In a normal situation, the variable corresponding to the `id` is already there in the symbol table. Its index (or pointer) to the symbol table and the type of it may be two attributes of E .

¹If necessary, it may store the value as well.

```

%union {
    .....
    struct {
        int stIndex;
        char typeCode;
    } eIndType;
    .....
}

```

5. If the variable is not declared, print an error message, but insert the variable in the symbol table with a special type tag 'UNDEF. Copy the index and type to *E*. This will ensure that the error message for an undeclared variable will not be generated more than once.
6. At the end of parsing print the content of the symbol table.
7. Prepare a Makefile for compilation. We have the following files (at the least): Makefile, <roll no>.1, <roll no>.y++, symtab.h, symTab.c++. Prepare a tar file with the name <roll no>.7.tar and send it.

Sample Input and Output:

```

// This program computes factorial
n fact i : int
read n
i = 1
fact = 1
while i < n | i == n do
    fact = fact * i
    i = i + 1
; // Note the ;
print fact

```

```

Accepted
int n Size: 4 Displacement: -12
int fact Size: 4 Displacement: -8
int i Size: 4 Displacement: -4

```

```

// Program with error
n n fact : int
a b b : real
read n
i = 1
fact = 1
while i < n | i == n do
    fact = fact * i
    i = i + 1
; // Note the ;
print fact

```

```

Re-declaration of variable: n at line no: 2
Re-declaration of variable: b at line no: 3
Variable not declared: i at line no: 7
Rejected
int n Size: 4 Displacement: -8
int fact Size: 4 Displacement: -4
float a Size: 4 Displacement: -16
float b Size: 4 Displacement: -12
undefined i

```