

### CD Assignment-3

Name: Sushanth M Nair

SRN: PES1UG20CS450

Section: H

Roll No: 50

Input:

```
sample_input1.c
~/Downloads/cd_lab_ass3

1 //PES1UG20CS450
2 int main()
3 {
4     int a=10;
5     float b=5;
6     double c=9.8;
7     char d;
8 }
```

Output:

```
1 //PES1UG20CS450
2 int main()
3 {
4     int a=10;
5     float b=5;
6     double c=9.8;
7     char d;
8 }
```

```
seed@VM: ~/.../cd_lab_ass3

[02/28/23] seed@VM: ~/.../cd_lab_ass3$ ./run.sh
[02/28/23] seed@VM: ~/.../cd_lab_ass3$ cat output1.txt
Valid syntax
Name      size  type  lineno  scope  value
a         2     2     3       1      10
b         4     3     4       1      5
c         8     4     5       1     9.8
d         1     1     6       1      ~
[02/28/23] seed@VM: ~/.../cd_lab_ass3$
```

.h file:

```
#define CHAR 1
#define INT 2
#define FLOAT 3
#define DOUBLE 4

typedef struct symbol          //data structure of items in
the list
{
    char* name;                //identifier name
    int size;                  //storage size of
identifier name
    int type;                  //identifier type
    char* val;                 //value of the
identifier
    int line;                  //declared line number
    int scope;                 //scope of the variable
    struct symbol* next;
}symbol;

typedef struct table           //keeps track of the start of
the list
{
    symbol* head;
}table;

static table* t;

table* init_table();           //allocate a new empty symbol table
symbol* init_symbol(char* name, int size, int type, int lineno,
int scope);                   //allocates space for items in the list
void insert_symbol(char* name, int size, int type, int lineno,
int scope);                   //inserts symbols into the table when
declared

void insert_val(char* name, char* v, int line);           //-
inserts values into the table when initialised
int check_sym_tab(char* name);                           //checks symbol
table whether the variable has been declared or not
void display_sym_tab();                                   //-
displays symbol table
int size(int type);                                       //gets the size of the
type(macros defined before)
```

.sh file:

```
#!/bin/bash
```

```
lex lexer.l
```

```
yacc -d parser.y -Wno
```

```
gcc -g y.tab.c lex.yy.c -ll
```

```
./a.out<sample_input1.c>output1.txt
```

```
~
```