COMPILER DESIGN REPORT

1)FlatB programming language description:

FlatB programming language is a simple language.A program in FlatB contains two blocks namely declbock,codeblock.All the variables have to be declared in the declblock before being used in the codeblock. FlatB programming language accepts the Data types Integers and Array of Integers .Multiple variables can be declared in the statement.Each declaration statement ends with a “;”.The codeblock contains statements of the type read,print, normal arithmetic equation statements,conditionals, loops, labels.

2)Syntax and Semantics:

A sample program in the FlatB programming language looks like

```
declblock{
        int data[100];
        int i, sum;
}

codeblock{
        for i = 1, 100 {
                data[i] = i ;
        }
L1:     sum = 0 ;
        for i = 1, 100 {
                sum = sum + data[i] ;
        }
Goto L1 if sum<10000;
```

```
        print "Sum value: ", sum ;
}
```

Grammar of the FlatB I:

As said earlier every program in FlatB language contains two blocks .All the strings in uppercase are the tokens of respective keywords.

```
program     :        decl_block code_block
```

Declaration block expands as follows:

```
        decl_block  : DECLBLOCK '{' decLIST '}'
                    ;

        decLIST   : declarationstatement   decLIST
                  | declarationstatement



         declarationstatement  : type var_decl remain_decl SMC
                               | type var_decl SMC
                               ;

        var_decl : IDENTIFIER
                 | IDENTIFIER '=' NUMBER
                 | IDENTIFIER '=' '{' NUMBER '}'
                 | IDENTIFIER '[' NUMBER ']'
                 | IDENTIFIER '=' '{' NUMBER remainarray '}'
                 ;

       remainarray  : COMA NUMBER remainarray
```

```
                        | COMA NUMBER

        remain_decl : COMA var_decl remain_decl
                       | COMA var_decl
                      ;



         type  : TYPE_INTEGER
               ;
```

After the declarations are finished codeblock is written.Grammar of the codeblock is as follows:

```
code_block:  CODEBLOCK '{' codeLIST '}'

codeLIST : codeLIST codedeclaration
           |codedeclaration
           ;




codedeclaration :

         READ read_var SMC

           |PRINT STRING COMA var SMC
           |PRINTL STRING SMC

           |WHILE boolexpr '{' codeLIST '}'

           |FOR IDENTIFIER '=' constant COMA constant '{' codeLIST '}'
            |FOR IDENTIFIER '=' constant COMA constant COMA
constant '{' codeLIST '}'
```

```
                |IF boolexpr '{' codeLIST '}'
                |IF boolexpr '{' codeLIST '}' ELSE boolexpr '{' codeLIST '}'

                |GOTO IDENTIFIER SMC
                |GOTO IDENTIFIER IF boolexpr SMC

                | IDENTIFIER ':'
                | var '=' mathexpr SMC
                | var '=' boolexpr SMC




var         :IDENTIFIER '[' mathexpr ']'
            |IDENTIFIER
            ;

read_var  :   IDENTIFIER
               | IDENTIFIER '[' IDENTIFIER ']'   /* array */
               ;




boolexpr : '(' boolexpr ')'
              |boolexpr OR boolexpr
              |boolexpr AND boolexpr
              |mathexpr GT mathexpr
              |mathexpr GE mathexpr
              |mathexpr LT mathexpr
              |mathexpr LE mathexpr
              |mathexpr EQ mathexpr
;
```

```
mathexpr     : '(' mathexpr ')'
             |mathexpr '+' mathexpr
             |mathexpr  '-' mathexpr
             |mathexpr '/'  mathexpr
             |mathexpr '*' mathexpr
             |constant
;
constant     : IDENTIFIER
             | IDENTIFIER '[' mathexpr ']'
             | NUMBER
;
```