COMPILER DESIGN LAB

WEEK 5 (22.1.19) – EXERCISE

SET A

1. Write a lex program to convert the following while statement to for statement.

```
while(condition) {
statement(s);
}
```

Input:

```
x = 0;
while (x < 3) {
      print x;
      x = x + 1;
      }
```

2. Convert the given switch case statement to else if statement.

```
switch (expression)
{
  case value1:
    statement1;
    break;
  case value2:
    statement2;
    break;
  default:
    statementDefault;
}
```

3. Write a lex program to convert the following nested for loop statement to nested do-while statement.

Input:

```
   for ( init; condition; increment ) {
      for ( init; condition; increment ) {
         statement(s);
      }
      statement(s);
   }
```

4. Write a lex program to convert the following nested if-else statement to single if-else statement.

```
if(x > y) {
      if(x > z)
            x is greater
      else
            x is not greater
      }
else
      x is not greater
```

COMPILER DESIGN LAB

WEEK 5 (22.1.19) – EXERCISE

SET B

1. Write a lex program to convert the following for statement to do-while statement.

```
for (<initial statement(s)>; <Condition expression>; <Repeat step(s)>)
     {
              <Loop statement(s)>;
            }
```

2. Convert the given if-else statement to switch case.

```
if (condition)
    statement;
else if (condition)
    statement;
.
.
else
    statement;
```

3. Write a lex program to convert the following nested do while statement to nested for loop statement.

Input:

```
do {
   statement(s);

   do {
      statement(s);
   }while( condition );

}while( condition );
```

4. Write a lex program to convert the following nested for statement to single for statement.