**1.Design a Scanner to identify the different type of tokens in the following Code segment and also print the total number of tokens.**

**#include <stdio.h>**

**// Sample procedure**

**void greetUser(char name[]) {**

**printf("Hello, %s! Welcome to the program.\n", name);**

**}**

**int main() {**

**char username[50];**

**printf("Enter your name: ");**

**scanf("%s", username);**

**// Call the procedure**

**greetUser(username);**

**printf("This is a simple C program.\n");**

**return 0;**

**}**

**2.Design a PARSER for the following Code Segment:**

**#include <stdio.h>**

**int main()**

**{**

**int choice;**

**printf("Enter a choice (1-3): ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**printf("You chose Option 1.\n");**

**break;**

**case 2:**

**printf("You chose Option 2.\n");**

**break;**

**case 3:**

**printf("You chose Option 3.\n");**

**break;**

**default:**

**printf("Invalid choice. Please enter a number between 1 and 3.\n");**

**}**

**return 0;**

**}**

**3. Design a PARSER for the following Code Segment:**

**using namespace std;**

**class Counter**

**{**

**private:**

**unsigned int count; //count**

**public:**

**Counter() : count(0) //constructor**

**{ }**

**unsigned int get\_count() //return count**

**{ return count; }**

**void operator ++ () //increment (prefix)**

**{**

**++count;**

**}**

**};**

**int main()**

**{**

**Counter c1, c2; //define and initialize**

**cout << “\nc1=” << c1.get\_count(); //display**

**cout << “\nc2=” << c2.get\_count();**

**++c1; //increment c1**

**++c2; //increment c2**

**++c2; //increment c2**

**cout << “\nc1=” << c1.get\_count(); //display again**

**cout << “\nc2=” << c2.get\_count() << endl;**

**return 0;**

**}**

**4.Design a Scanner to identify the different type of tokens in the following Code segment and also print the total number of tokens.**

**#include <stdio.h>**

**int main() {**

**char input[100]; // Assuming a maximum input length of 100 characters**

**while (1) {**

**printf("Enter input (type 'exit' to quit): ");**

**fgets(input, sizeof(input), stdin);**

**// Remove newline character from the input**

**int len = strlen(input);**

**if (len > 0 && input[len - 1] == '\n') {**

**input[len - 1] = '\0';**

**}**

**// Check for the exit condition**

**if (strcmp(input, "exit") == 0) {**

**printf("Exiting the loop.\n");**

**break;**

**}**

**// Process the input or perform any desired operations**

**printf("You entered: %s\n", input);**

**}**

**return 0;**

**}**

**5. Write a program using Lex and Yacc to parse and execute a program that contains both declarative statements and ‘for’ loop and ‘if’constructs.**

**#include<iostream>**

**using namespace std;**

**int main() {**

**string l;**

**getline(cin, l, '$');**

**for (int i = 0; i < l.length() - 1; i++) {**

**if (l[i] == ' ' && l[i + 1] != ' ') {**

**l.insert(i + 1, " ");**

**i++; // Skip the added space**

**}**

**}**

**cout << l << endl;**

**return 0;**

**}**

**6. Implement a Program to Accept a Context Free Grammar and to print the First set in Top Down Parsing using C or Java or Python.**