Compiler Design Lab Assignments
Subject Code: CS793

1. Write a program to recognize identifiers.
   Output:
   Input a string: compiler It
   is an identifier
   Input a string: 123_abc
   It is not an identifier

2. Write a program in C to print the number of tokens in another C program(input a string).
   Output:
   Input a string: int a=10;

   The tokens are: int, a, =, 10, ;

3. Write a C program that will replace a malloc() function to calloc() function with appropriate parameters.(E.g.- Input: malloc(2*sizeof(int)), Output: calloc(2, sizeof(int)))
   N.B.- An user defined malloc function should implement the corresponding calloc calling.

4. Write a C program to check if another C file has the necessary prerequisites for executing a C program(such as #include<stdio.h>, int main(){},return 0;)
   Output:
   #include<stdio.h>
   int main()
   {int a=10;return 0;}

   Valid program

   #include<stdio.h>
   int main()
   {int a=10;}

   No return statement

5. Write a program to recognize constants.
   Output:
   Input a program: #include<stdio.h>
   Void main()

   {const int a=10;
   Int b;} Constant
   : a Void main()

   {int a;}

Constant : not found

6. Write a C program to find the number of alphabets, digits and special characters in a given input.
7. Write a C program which will take a C file as input and find out the total memory space required for allocating all the variables and check if it exceeds a certain limit.(which is taken as user input)
   Output:
   int a,b,c;
   float x,y;
   Total memory space required = (2*3)+(2*4)=14
   N.B.-Please note sizeof operator must be used
8. Write a C program that takes another C file as input and check if all the functions are having proper return types. (Hint: Check whether the return type matches with the return statement.)
   Output:
   int fun()
   {return 0;}

   Function fun() have proper return types

   float fun()
   {int b=6;
   return b;}
   Function fun() does not have proper return type

9. Write a program to recognize keyword.
   Output:
   Input a program:
   Void main()
   {int a=;}
   Keywords: void, int

10. Write a C program that will take another C file and check if all the functions defined in it are called or not. If not, it just delete the function.
    Output:
    Input a program:
    Void fun1(){printf("hi..");}
    Void fun2(){printf("this function is not called");
    Void main()
    {fun1();
    fun2();}

    Void fun1(){printf("hi..");}
    Void main()
    {fun1();}

11. Write a C program that should check if all the members of the structures are having a defined data type. If not, print an error.
    Output:
    Input a structure definition:
    Struct demo{
    Int a;
    Float b;
    };
    The structure demo has defined data types

    Input a structure definition:
    Struct demo{
    class a;
    Float b;
    };
    Class is undefined

12. Write a program to ignore the comments in the given input source program(i.e., delete them).
    Output:
    Input a program:
    Void main()
    {int a;
    //this is a comment line
    float b;
    }

    Void main()
    {int a;
    float b;
    }

13. Write a C program that will check whether the input string is containing “Monday” in it.
14. Write a C program that will take a C file as an input and output a file which will have \n and \b replaced by corresponding spaces.
    Output:
    Input a file:
    #include<stdio.h>
    Void main()
    {printf(“hi students\t this is compiler design”);
    printf(“uem\nkolkata”);}

    #include<stdio.h>
    Void main()

{printf(“hi students this is compiler design”);
printf(“uem
kolkata”);}

15. Write a C program that will check whether all the variables declared in an input file are initialized or not. If not, initialize them with 0.
    Output:
    void main()
    {int a,b=10,c;} After
    initialization:

    void main()

    {int a=0,b=10,c=0;}

16. Write a C program to identify whether a given line is a comment or not.
    Output:
    Input a program: #include<stdio.h>
    Void main()

    {

    int a;

    //this is a comment
    float b;

    /*double d;
    printf(“Hello”);*/
    printf(“Hi..”);

    }

    Comment: this is a comment
    Comment: double d; printf(“Hello”);

17. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
18. Write a C program that will look up a dictionary (a file may be) and autocorrect a word taken from the user if two letters are erroneous.
    Output:
    Input the words in dictionary: hello, uem, exam
    Input a word: heo

    Autocorrect suggests: hello

19. Write a C program to simulate lexical analyzer for validating operators.
    Output:
    Input a program:
    #include<stdio.h>
    Void main()

    {int a, b=10,c=20;
    a=b+c;}

    Valid operators: {=, =, =, +}

    Input a program:
    #include<stdio.h>
    Void main()

    {int a, b=10,c=20;

    50=a;}

    Valid operators: {=,=}

20. Write a C program to replace all the digits in a file to their corresponding words. Use a switch case.
21. Write a program in C that will take two files as input and merge them into one and delete any redundant words from the resulting file.

22. Write a C program to find if a given grammar is Context free or not
    Output:
    Input the productions of a grammar:

    S->aAb

    A->b

    It is context free

    aAb->aab

    It is not context free

23. Write a C program to count the number of white spaces between two consecutive tokens in a program and replace it with a single whitespace.
    Output:
    Enter a program:

    #include<stdio.h>

    Void main()

{int a= 20;float b=30.0;}
The modified program is:
#include<stdio.h>

Void main()

{int a=20;float b=30.0;}

24. Write a program in C to find the First and Follow for a given set of productions
Output:
Enter the no. of Non-terminals in the grammer:3
Enter the Non-terminals in the grammer: E T V
Enter the no. of Terminals in the grammer: 5 Enter
the Terminals in the grammer: + * ( ) i

Enter the production for E (End the production with '$' sign) :(i)$

Enter the production for T (End the production with '$' sign) :i*E$ Enter the production for V (End the production with '$' sign) :E+i$ The production for E -> (i)

The production for T -> i*E
The production for V -> E+i
The first of E -> (

The first of T -> i
The first of V -> (

25. Write a program in C to simulate #define on a given file.
Output:
Enter a program: #include<stdio.h>
#define int float

Void main()

{int a;}

The modified program is:
#include<stdio.h>

Void main()

{float a;}

26. Write a C program to find if a given grammar is Context sensitive or not
Output:
Input the production rules:
aSb->atb

The grammar is context-sensitive

abSt->df

The grammar is not context sensitive

27. Write a program in C which can recognize a web address.
    Output:
    Enter a string: https://www.uem.edu.in Valid
    web address

    Enter a string: https:www.uem.in Not
    valid web address

28. Write a C program to simulate the working of #include<some_file_name>. (some_file_name should be the input and it's contents must be copied and pasted in another file along with your program code).
29. Write a C program to convert all the lowercase alphabets in a file to uppercase and vice-versa.

30. Write a C program to find if a given grammar is Regular or not
    Output:
    S->a

    S->z
    T->aS

    Grammar is regular
    S->Ab

    A->Ba
    B->c

    Grammar is not regular

31. Write a C program which can recognize whether an email address is valid or not.
    Output:
    uem@gmail.com
    Email is valid
    Uem#gmail.com
    Email is not valid
32. Given a set of alphabets {a,b,c}, write a C program to find all the possible strings of length at most 3.
    Output:
    This is uem university of engineering and management At
    most 3: is, uem, of, and

33. Write a C program to find out if there is a dangling else in a program.
    Output:
    Enter a program:
    If(condition)

```
If(condition)
        Printf("hi..");
Else
        Printf("hello");
```

Dangling else found

Enter a program:

```
If(condition)
        Printf("hi..");
Else
        Printf("hello");
```

No dangling else found

34. Write a C program to find the number of new lines in a program
    Output:
    "This is uem\nAll the best\nExecute"
    New lines: 3

35. Write a C program to print the number of times a certain function is called.
    Output:

```
Void fun()
{Printf("hi!!");}
Void main()
{fun();
printf("Hello");
fun();}
```

    Number of times called: fun()->2

36. Write a C program which will copy each line of a given program and number each newline.
    Output:

```
#include<stdio.h>
Void main()
{
Printf("hi");
}
```

    Line 1: #include<stdio.h>
    Line 2: Void main()

Line 3:{

Line 4: Printf(“hi”);

Line 5: }

37. Write a program in C to print an error when a user doesn’t provide a semi-colon at the end of a line of a program. User input should be a file containing a program.(N.B.- For loop doesn’t have a semicolon at the end normally)

38. Write a C program to implement strcat() function.
    Output:
    Input first string: hello Input
    second string: world

    After concatenation: helloworld

39. Write a program in C to count the number of blank spaces and print the number of lines.
    Output:
    Enter a string: int a=10; \nfloat b=20;\nreturn 0;
    Blank spaces = 4

    Number of lines = 3

40. Write a C program to count the number of times a variable’s value is updated throughout the execution of the program.
    Output:
    Enter a string: int a=10,b=5; a++; a=a-7; printf(“%d”,a);
    Updated: a->3, b->1

41. Write a C program that will count the number of lowercase and uppercase characters from a file.