SSGMCE	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a C program to test whether a given identifier is valid or not.	
EXPERIMENT NO. : **SSGMCE/WI/IT/01/6IT01/03**	ISSUE NO. : 00	ISSUE DATE : 01.02.2022
REV. DATE :	REV. NO. :	DEPTT. : INFORMATION TECHNOLOGY
LABORATORY : COMPILER DESIGN (CD)	SEMESTER : VI	PAGE: OF 4

1.0) AIM:

Write a C program to test whether a given identifier is valid or not.

2.0) OBJECTIVE:

After the completion of this experiment, lexical analyzer will be able to identify whether a given identifier is valid or not.

3.0) FACILITIES/ APPARATUS:

i) Hardware : Computer Machine

ii) Software : Turbo C++

4.0) THEORY:

C identifiers represent the name in the C program, for example, variables, functions, arrays, structures, unions, labels, etc. An identifier can be composed of letters such as uppercase, lowercase letters, underscore, digits, but the starting letter should be either an alphabet or an underscore. If the identifier is not used in the external linkage, then it is called as an internal identifier. If the identifier is used in the external linkage, then it is called as an external identifier.

Rules for constructing C identifiers

1. The first character of an identifier should be either an alphabet or an underscore, and then it can be followed by any of the character, digit, or underscore.
2. In identifiers, both uppercase and lowercase letters are distinct, i.e., identifiers are case sensitive.
3. Commas or blank spaces cannot be specified within an identifier.
4. Keywords cannot be represented as an identifier.
5. The length of the identifiers should not be more than 31 characters.
6. Identifiers should be written in such a way that it is meaningful, short, and easy to read.

5.0) PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
```

```
{
        clrscr();
        char string[25];
        int count=0, flag;
        printf("Enter any string for Identifier: ");
        gets(string);
        if((string[0]>='a' && string[0]<='z') || (string[0]>='A' && string[0]<='Z') || (string[0]=='_'))
        {
                for(int i=1; i<=strlen(string); i++)
                {
                        if((string[i] >= 'a' && string[i] <= 'z') || (string[i] >= 'A' && string[i] <= 'Z') ||
                        (string[i] >= '0' && string[i] <= '9') || (string[i] == '-'))
                        {
                                count++;
                        }
                }
                if(count == strlen(string))
                {
                        flag=0;
                }
        }
        else
        {
                flag=1;
        }
        if(flag == 1)
                printf("%s is not valid identifier", string);
        else
                printf("%s is valid identifier", string);
        getch();
}
```

6.0) OUTPUT OF PROGRAM

Enter any string for Identifier:

INPUT

ssgmce123

OUTPUT

ssgmce123 is valid identifier

Enter any string for Identifier:

INPUT

123ssgmce

OUTPUT

123ssgmce is not valid identifier

7.0) CONCLUSION:

A lexical analyzer has been designed using C language for the given language in which it identifies the valid identifiers.