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

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

# Algorithm:

* 1. Start
  2. Read the given input string.
  3. Check the initial character of the string is numerical or any special character except ‘\_’ then print it is not a valid identifier.
  4. Otherwise print it as valid identifier if remaining characters of string doesn’t contains any special characters except ‘\_’.
  5. Stop.

# Program:

#include<stdio.h> #include<conio.h> #include<ctype.h>

void main()

{

char a[10]; int flag, i=1; clrscr();

printf("\n Enter an identifier:"); gets(a);

if(isalpha(a[0]))

flag=1; else

printf("\n Not a valid identifier");

while(a[i]!='\0')

{

if(!isdigit(a[i])&&!isalpha(a[i]))

{

flag=0; break;

} i++;

}

if(flag==1)

printf("\n Valid identifier"); getch();

}

**Sample Input & Output: Input**: Enter an identifier: first ***Output:***

Valid identifier

Enter an identifier:1aqw Not a valid identifier

2. Write a C program to identify whether a given line is a comment or not

**Aim:** C program to identify whether a given line is a comment or not

# Program:

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

{

char com[30]; int i=2,a=0; clrscr();

printf("\n Enter comment:");

gets(com); if(com[0]=='/') {

if(com[1]=='/')

printf("\n It is a comment"); else if(com[1]=='\*') {

for(i=2;i<=30;i++)

{

if(com[i]=='\*'&&com[i+1]=='/')

{

printf("\n It is a comment"); a=1; break; }

else continue; } if(a==0)

printf("\n It is not a comment");

}

else

else

printf("\n It is not a comment");

}

printf("\n It is not a comment"); getch();

}

**Sample Input & Output: Input:** Enter comment: //hello **Output**: It is a comment **Input:** Enter comment: hello **Output**: It is not a comment

**3.Write a C program to validate operators.**

Aim: Write a C program to validate operators.

Algorithm:

1. Start

2. Read the given input.

3. If the given input matches with any operator symbol.

4. Then display in terms of words of the particular symbol. Else print not an operator.

5. Stop

Program:

#include<stdio.h>

#include<conio.h>

void main()

{

char s[5];

clrscr();

printf("\n Enter any operator:");

gets(s);

switch(s[0])

{

case'>': if(s[1]=='=')

printf("\n Greater than or equal");

else

printf("\n Greater than");

break;

case'<': if(s[1]=='=')

printf("\n Less than or equal");

else

printf("\nLess than");

break;

case'=': if(s[1]=='=')

printf("\nEqual to");

else

printf("\nAssignment");

break;

case'!': if(s[1]=='=')

printf("\nNot Equal");

else

printf("\n Bit Not");

break;

case'&': if(s[1]=='&')

printf("\nLogical AND");

else

printf("\n Bitwise AND");

break;

case'|': if(s[1]=='|')

printf("\nLogical OR");

else

printf("\nBitwise OR");

break;

case'+': printf("\n Addition");

break;

case'-': printf("\nSubstraction");

break;

case'\*': printf("\nMultiplication");

break;

case'/': printf("\nDivision");

break;

case'%': printf("Modulus");

break;

default: printf("\n Not a operator");

}

getch();

}

**Sample Input & Output:**

Input

Enter any operator: \*

Output Multiplication

**4. Develop a lexical analyzer to identify identifiers, constants, operators using C program.**

#include<stdio.h>

#include<ctype.h>

#include<string.h>

void main(){

int i,ic=0,m,cc=0,oc=0,j;

char b[30],operators[30],identifiers[30],constants[30];

printf("enter the string : ");

scanf("%[^\n]s",&b);

for(i=0;i<strlen(b);i++){

if(isspace(b[i])){

continue;

}

else if(isalpha(b[i])){

identifiers[ic] =b[i];

ic++;

}

else if(isdigit(b[i])){

m=(b[i]-'0');

i=i+1;

while(isdigit(b[i])){

m=m\*10 + (b[i]-'0');

i++;

}

i=i-1;

constants[cc]=m;

cc++;

}

else{

if(b[i]=='\*'){

operators[oc]='\*';

oc++;

}

else if(b[i]=='-'){

operators[oc]='-';

oc++;

}

else if(b[i]=='+'){

operators[oc]='+';

oc++;

}

else if(b[i]=='='){

operators[oc]='=';

oc++;

}

}

}

// printing

printf(" identifiers : ");

for(j=0;j<ic;j++){

printf("%c ",identifiers[j]);

}

printf("\n constants : ");

for(j=0;j<cc;j++){

printf("%d ",constants[j]);

}

printf("\n operators : ");

for(j=0;j<oc;j++){

printf("%c ",operators[j]);

}

}

1. **Write a C program to implement symbol table.**

include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<conio.h> int cnt=0;

struct symtab

{

char label[20]; int addr;

}sy[50]; void insert();

int search(char \*);

void display();

void modify();

void main()

{

int ch,val; char lab[10];

do

{

printf("\n 1.insert\n2.display\n3.search\n4.modify\n5.exit\n"); scanf("%d",&ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

display();

break;

case 3:

printf("enter the label");

scanf("%s",lab);

val=search(lab);

if(val==1)

printf("label is found");

else

printf("label is not found");

break;

case 4:

modify();

break;

case 5:

exit(0);

break;

}

}while(ch<5);

}

void insert()

{

int val;

char lab[10]; printf("enter the label"); scanf("%s",lab); val=search(lab); if(val==1)

printf("duplicate symbol");

else

{

strcpy(sy[cnt].label,lab);

printf("enter the address");

scanf("%d",&sy[cnt].addr);

cnt++;

}

}

int search(char \*s)

{

int flag=0,i; for(i=0;i<cnt;i++)

{

if(strcmp(sy[i].label,s)==0)

flag=1;

}

return flag;

}

void modify()

{

int val,ad,i;

char lab[10];

printf("enter the label"); scanf("%s",lab); val=search(lab); if(val==0)

printf("no such symbol");

else

{

printf("label is found \n");

printf("enter the address");

scanf("%d",&ad);

for(i=0;i<cnt;i++)

{

if(strcmp(sy[i].label,lab)==0)

sy[i].addr=ad;

}

}

}

void display()

{

int i;

for(i=0;i<cnt;i++)

printf("%s\t%d\n",sy[i].label,sy[i].addr);

}