AIM :- Implementation of scanner phase (w/o using any automated tools).

PROCEDURE :-

- Write scanner algorithm for implementing the “D” language
- Generate scanner in “C” code for language “D” given above

Program:

D_lang_C_prog.c

```
#include<string.h>
#include<stdio.h>
void identify(char s[],int len)
{
   int flag=0;
   int i=0;
          while(i<len)
   {
      flag=0;
      if((int)s[i]>=48 && (int)s[i]<=57)
      {
         while((int)s[i]>=48 && (int)s[i]<=57)
         {
            i++;
         }
         if(s[i]=='.')
         {
                                          i++;
            while((int)s[i]>=48 && (int)s[i]<=57)
            {
               i++;
            }
            printf("float \n");
         }
         else
         {
```

```
        printf("integer \n");
    }
}
else if(s[i]=='.')
{
    i++;
    while((int)s[i]>=48 && (int)s[i]<=57)
    {
        i++;
    }
    printf("float \n");
}
else if(s[i]=='i')
{
    i++;
    if(s[i]=='f')
    {
        printf("keyword: if\n");
        i+=1;
    }
    else
    {
        i--;
    }
}
else if(s[i]=='f')
{
    i++;
    if(s[i]=='o' && s[i+1]=='r')
    {
        printf("keyword: for\n");
        i+=2;
```

```
    }
    else
    {
        i--;
    }
}
else if(s[i]=='w')
{
    i++;
    if(s[i]=='h' && s[i+1]=='i' && s[i+2]=='l' && s[i+3]=='e')
    {
        printf("keyword: while\n");
        i+=4;
    }
    else
    {
        i--;
    }
}
else if(s[i]=='d')
{
    i++;
    if(s[i]=='o')
    {
        printf("keyword: do\n");
        i+=1;
    }
    else
    {
        i--;
    }
}
```

```
else if(s[i]=='e')
{
   i++;
   if(s[i]=='l')
   {
      i++;
      if(s[i]=='s' && s[i+1]=='e')
      {
          printf("keyword: else\n");
          i+=2;
      }
      else
      {
         i--;
      }
   }
   else if(s[i]=='x')
   {
      i++;
      if(s[i]=='i' && s[i+1]=='t')
      {
          printf("keyword: else\n");
          i+=2;
      }
      else
      {
         i--;
      }
   }
   else
   {
         i--;
```

```
    }
}
else if(s[i]=='c')
{
    i++;
    if(s[i]=='a' && s[i+1]=='s' && s[i+2]=='e')
    {
        printf("keyword: case\n");
        i+=3;
    }
    else
    {
       i--;
    }
}
else if(s[i]=='s')
{
    i++;
    if(s[i]=='w' && s[i+1]=='i' && s[i+2]=='t' && s[i+3]=='c' && s[i+4]=='h')
     {
        printf("keyword: switch\n");
        i+=5;
     }
    else
    {
       i--;
    }
}
else if(s[i]=='u')
{
    i++;
    if(s[i]=='n' && s[i+1]=='t' && s[i+2]=='i' && s[i+3]=='l')
```

```
        {
            printf("keyword: until\n");
            i+=4;
        }
        else
        {
            i--;
        }
    }
    else if(((int)s[i]>=65 && (int)s[i]<=90) || ((int)s[i]>=97 && (int)s[i]<=122))
    {

        while(((int)s[i]>=65 && (int)s[i]<=90) || ((int)s[i]>=97 && (int)s[i]<=122) || ((int)s[i]>=48 &&
(int)s[i]<=57)  )
        {
            i++;
        }
        printf("identifier\n");
    }
    else if(s[i]=='[')
        {printf("Punctuation: [\n");i++;}

    else if(s[i]==']')
        {printf("Punctuation: ]\n");i++;}

    else if(s[i]=='{')
        {printf("Punctuation: {\n");i++;}

    else if(s[i]=='}')
        {printf("Punctuation: }\n");i++;}

    else if(s[i]=='(')
        {printf("Punctuation: (\n");i++;}
```

```
else if(s[i]==')')
   {printf("Punctuation: )\n");i++;}
else if(s[i]==',')
   {printf("Punctuation: ,\n");i++;}

else if(s[i]=='+')
{
   printf("Operator: +\n");
   i++;
}
else if(s[i]=='-')
{
   printf("Operator: -\n");
   i++;
}
else if(s[i]=='*')
{
                        i++;
                        if(s[i]!='*')
                        {
                                printf("Operator: *\n");
                        }
                        else if(s[i]=='*')
                        {
                           i++;
                                if(s[i]=='*')
                                {
                                        while(i<len)
                                        {
                                                i++;
                                        }
```

```
                                        printf("comment \n");
                                    }
                                }
                            }
else if(s[i]=='%')
{
   printf("Operator: %%\n");
   i++;
}
else if(s[i]=='<')
{
   i++;
   if(s[i]=='>')
   {
         printf("Operator: <>\n");
         i++;
   }
   else if(s[i]=='=')
   {
      printf("Operator: <=\n");
      i++;
   }
   else
   {
      i--;
   }
}
else if(s[i]=='=')
{
   i++;
   if(s[i]=='=')
   {
```

```
        printf("Operator: ==\n");
        i++;
    }
    else
    {
        printf("Operator: =\n");
    }
}
else if(s[i]=='>')
{
    i++;
    if(s[i]=='=')
    {
        printf("Operator: >=\n");
        i++;
    }
    else
    {
        i--;
    }
}
else if(s[i]=='!')
{
    i++;
    if(s[i]=='=')
    {
        printf("Operator: !=\n");
        i++;
    }
    else
    {
        i--;
```

```
    }
}
else if(s[i]=='>')
{
    i++;
    if(s[i]=='=')
    {
        printf("Operator: >=\n");
        i++;
    }
    else
    {
        i--;
    }
}
else if(s[i]=='\'')
{
    i++;
    while(s[i]!='\'')
    {
        i++;
        if(i>len)
        {
            flag=1;
            break;
        }
    }
                                i++;
    if(flag==0)
    printf("String \n");
}
                    else if((int)s[i]==32)
```

```
                {
                        i++;
                }
                else if(s[i]=='\t')
    {
                        i+=4;
    }
    else
    {
       printf("unrecognised: %c",s[i]);
       i++;
    }
  }
}
int main()
{
   char name[] = " " ;
   gets(name);
        identify(name,strlen(name));
        return 0;
}
```

Output:-

```
main.c:332:5: warning: 'gets' is deprecated [-Wdeprecated-declarations]
/usr/include/stdio.h:638:14: note: declared here
main.c:(.text+0xab5): warning: the `gets' function is dangerous and should not be used.
printf("hello")
identifier
Punctuation: (
unrecognised: "identifier
unrecognised: "Punctuation: )
Segmentation fault (core dumped)


...Program finished with exit code 139
Press ENTER to exit console.
```