1. **Detect whether the given input string is keyword or not:**

|  |
| --- |
| //program 1: detect keyword or not keyword  #include <stdio.h>  #include <string.h>  #define MAX 5  #define max() {}  int main() {  char keyword[32][10]={  "auto","double","int","struct","break","else","long",  "switch","case","enum","register","typedef","char",  "extern","return","union","const","float","short",  "unsigned","continue","for","signed","void","default",  "goto","sizeof","voltile","do","if","static","while"  } ;  char str[20];  puts("Enter a string");  gets(str);  int flag=0,i;  for(i = 0; i < 32; i++) {  if(strcmp(str,keyword[i])==0) {  flag=1;  }  }  if(flag==1)  printf("%s is a keyword",str);  else  printf("%s is not a keyword",str);  } |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **keyword** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | **a** | **u** | **t** | **o** | **\0** |  |  |  |  |  |
| **1** | **d** | **o** | **u** | **b** | **l** | **e** | **\0** |  |  |  |
| **2** |  |  |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **31** |  |  |  |  |  |  |  |  |  |  |

1. A sample program is given below that splits the contents of a .text file using space(‘ ‘):

|  |  |
| --- | --- |
| #include <iostream>  #include <fstream>  #include <string>  #include <sstream>  using namespace std;  int main() {  ifstream input("split\_text.txt"); string line;  while (getline(input, line, ' ')) {  cout << line << '\n';  }  return 0;  } | **split\_text.txt**  Hello World! I am CPP.  position = initial + rate \* 60 |

Sample output:

|  |
| --- |
|  |

1. Here are some practice problem of fseek()

|  |  |
| --- | --- |
| #include <stdio.h>  int main () {  FILE \*fp;  fp = fopen("file.txt","w");  fputs("This is tutorialspoint.com", fp);  fseek( fp, 0, SEEK\_SET );  fputs("C Programming", fp);  fclose(fp);  return(0);  } | Output:  **file.txt**  C Programmingialspoint.com |

A simple change of the above program:

|  |  |
| --- | --- |
| #include <stdio.h>  int main () {  FILE \*fp;  fp = fopen("file.txt","w");  fputs("This is tutorialspoint.com", fp);  fseek( fp, 14, SEEK\_SET );  fputs("C Programming", fp);  fclose(fp);  return(0);  } | Output:  **file.txt**  This is tutoriC Programming |

1. A program that reads a text file and identify the vowels and consonants:

Input.txt

|  |
| --- |
| Hello, World!  This is your first program in this lab. |

programFile.cpp

|  |
| --- |
| #include <fstream>  int main()  {  FILE \*fp;  int vowel=0,consonant=0;  char ch;  char message[200];  fp=fopen("input.txt","r");  if(fp==NULL)  {  printf("Source can't be opened");  exit(-1);  }  while(!feof(fp))  {  fgets(message, 200, fp);  printf("%s", message);  }  printf("\n\n");  fseek(fp, 0, SEEK\_SET);  while(ch!=EOF)  {  ch=fgetc(fp);  if((ch=='a')||(ch=='A')||(ch=='e')||(ch=='E')||(ch=='i')||(ch=='I')||(ch=='o') ||(ch=='O')||(ch=='u')||(ch=='U'))  {  vowel++;  printf("(%c) ", ch);  }  else if((ch>=65&&ch<=90)||(ch>=97&&ch<=122))  {  consonant++;  printf("%c ", ch);  }  }  fclose(fp);  printf("\n\nNumber of vowels are = %d\nNumber of consonants are = %d",vowel,consonant);  return 0;  } |

Output

|  |
| --- |
|  |