Assignment - 19

A Job Ready Bootcamp in C++, DSA and IOT MySirG

Handling multiple Strings in C Language

1. Write a program to find the number of vowels in each of the 5 strings stored in two dimensional arrays, taken from the user.

#include <stdio.h>

#include <string.h>

void countVowel(char str[])

{

int i=0,c=0;

while(str[i])

{

if(str[i]=='A'||str[i]=='E'||str[i]=='E'||str[i]=='I'||str[i]=='O'||str[i]=='U'||str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u')

c++;

i++;

}

printf("Number of vowels in %s is %d\n",str,c);

}

int main()

{

int i;

char str[5][20];

for (i = 0; i < 5; i++)

{

printf("Enter the word in %d string:\n", i + 1);

fgets(str[i], 20, stdin);

}

for (i = 0; i < 5; i++)

{

countVowel(str[i]);

}

printf("\n");

return 0;

}

2. Write a program to sort 10 city names stored in two dimensional arrays, taken from the user.

#include <stdio.h>

#include <string.h>

void sort\_city(char city[][20], int n)

{

for (int i = 0; i < n - 1; i++)

{

for (int j = i + 1; j < n; j++)

{

if (strcmp(city[j], city[i]) < 0)

{

char temp[20];

strcpy(temp, city[i]);

strcpy(city[i], city[j]);

strcpy(city[j], temp);

}

}

}

}

int main()

{

int i;

char city[10][20];

for (i = 0; i < 10; i++)

{

printf("Enter a city name:\n");

fgets(city[i], 20, stdin);

city[i][strcspn(city[i], "\n")] = '\0';

}

printf("Before sorting:\n");

for (int i = 0; i < 10; i++)

printf("%s\n", city[i]);

sort\_city(city, 10);

printf("after soring\n");

for (int i = 0; i < 10; i++)

printf("%s\n", city[i]);

printf("\n");

return 0;

}

3. Write a program to read and display a 2D array of strings in C language.

#include <stdio.h>

#include <string.h>

void display(int n, int m, char str[][m][20])

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

printf("%s ", str[i][j]);

}

printf("\n");

}

}

void input(char str[20])

{

fflush(stdin);

fgets(str, 20, stdin);

str[strcspn(str, "\n")] = '\0';

}

int main()

{

int n, m;

printf("Enter the size of 2d array of strings:\n");

scanf("%d %d", &n, &m);

char str[n][m][20];

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

printf("Enter string at row %d, column %d: ", i + 1, j + 1);

input(str[i][j]);

}

}

printf("Here is the 2D array of strings:\n");

display(n, m, str);

printf("\n");

return 0;

}

4. Write a program to search a string in the list of strings.

#include <stdio.h>

#include <string.h>

void search(int n, char str[], char list[][20])

{

int r = 1;

for (int i = 0; i < n; i++)

{

r = strcmp(str, list[i]);

if (r == 0){

printf("The string is present in the list in %d position\n", i);

return;

}

}

printf("The string is not present in the list \n");

}

void display(int n, char str[][20])

{

for (int i = 0; i < n; i++)

{

printf("%s ", str[i]);

printf("\n");

}

}

void input(char str[20])

{

fflush(stdin);

fgets(str, 20, stdin);

str[strlen(str) - 1] = '\0';

}

int main()

{

int static n, m;

printf("Enter the size of the list\n");

scanf("%d", &n);

char list[n][20];

for (int i = 0; i < n; i++)

{

printf("Enter string %d: ", i + 1);

input(list[i]);

printf("\n");

}

display(n, list);

char str[20];

printf("Enter the string you want to serach:\n");

input(str);

search(n, str, list);

printf("\n");

return 0;

}

5. Suppose we have a list of email addresses, check whether all email addresses have ‘@’ in it. Print the odd email out.

#include <stdio.h>

#include <string.h>

void input(int i, char str[][20])

{

fflush(stdin);

fgets(str[i], 20, stdin);

str[i][strlen(str[i])-1]='\0';

}

void display(int n, char str[][20])

{

for (int i = 0; i < n; i++)

printf("%d number email is %s\n", i + 1, str[i]);

}

void check(int n, char str[][20])

{

for (int i = 0; i < n; i++)

{

char \*pos1=strchr(str[i],'@');

char \*pos2=strrchr(str[i],'@');

if (pos1!=pos2)

{

printf("%d Number e-mail is not a valid email\n", i + 1);

}

}

}

int main()

{

int n;

printf("Enter the number of e-mails in the list:\n");

scanf("%d", &n);

char str[n][20];

printf("Enter the e-mails:\n");

for (int i = 0; i < n; i++)

{

input(i, str);

}

display(n, str);

check(n, str);

printf("\n");

return 0;

}

6. Write a program to print the strings which are palindrome in the list of strings.

#include <stdio.h>

#include <string.h>

void input(int i, char str[i][20])

{

fflush(stdin);

fgets(str[i], 20, stdin);

str[i][strcspn(str[i],"\n")]='\0';

}

void display(int n, char str[n][20])

{

for (int i = 0; i < n; i++)

{

printf("%d number string is :%s\n", i + 1, str[i]);

}

}

void check\_Palindrome(int i, char str[i][20])

{

int f = 0, len = strlen(str[i]);

for (int x = 0; x < len/2; x++)

{

if (str[i][x] != str[i][len -1- x])

{

f=1;

}

}

if (f)

printf("%s is not a palindrome\n", str[i]);

else

printf("%s is a palindrome\n", str[i]);

}

int main()

{

int n;

printf("Enter the number of string:\n");

scanf("%d", &n);

char str[n][20];

printf("Enter strings:\n");

for (int i = 0; i < n; i++)

{

input(i, str);

}

display(n, str);

for (int i = 0; i < n; i++)

{

check\_Palindrome(i, str);

}

printf("\n");

return 0;

}

7. From the list of IP addresses, check whether all ip addresses are valid.

#include <stdio.h>

#include <string.h>

#include<stdlib.h>

void input(char add[])

{

fflush(stdin);

fgets(add, 13, stdin);

add[strlen(add) - 1] = '\0';

}

void display(int n, char add[][13])

{

for (int i = 0; i < n; i++)

{

printf("%d number ip is %s\n", i + 1, add[i]);

}

}

void checkIP(int n, char ipadd[][13])

{

for (int i = 0; i < n; i++)

{

int f=0;

char \*a = strtok(ipadd[i], ".");

while (a != NULL)

{

int x = atoi(a);

if ((x >= 0) && (x <= 255))

f++;

a=strtok(NULL, ".");

}

if(f==4)

printf("The ip address %d is valid\n",(i+1));

else

printf("The ip adress %d is invalid\n",(i+1));

}

}

int main()

{

int n;

printf("please input the number of ip addresses:");

scanf("%d", &n);

char ipadd[n][13];

for (int i = 0; i < n; i++)

{

printf("Enter a ip Address:\n");

input(ipadd[i]);

}

display(n, ipadd);

checkIP(n, ipadd);

return 0;

}

8. Given a list of words followed by two words, the task is to find the minimum distance between the given two words in the list of words. (Example : s = {“the”,”quick”,”brown”,”fox”,”quick”} word1 = “the”, word2 = “fox”, OUTPUT : 1 )

#include <stdio.h>

#include <string.h>

#include <math.h>

void input(char WordList[15])

{

fflush(stdin);

fgets(WordList, 15, stdin);

WordList[strlen(WordList) - 1] = '\0';

}

void display(int n, char WordList[n][15])

{

for (int i = 0; i < n; i++)

{

printf("%d number word is %s\n", i + 1, WordList[i]);

}

}

void checkpos(int n, char word1[], char word2[], char WordList[][15])

{

int p1, p2;

for (int i = 0; i < n; i++)

{

if (strcmp(word1, WordList[i]) == 0)

p1 = i;

if (strcmp(word2, WordList[i]) == 0)

p2 = i;

}

if ((n-1 - abs(p1 - p2)) > abs(p1 - p2)-1)

printf("Distance between two words is %d\n", abs(p1 - p2)-1);

else

printf("Distnace between two given words is %d \n",(n-1-abs(p1-p2)));

}

int main()

{

int n;

printf("please input the number of Words:\n");

scanf("%d", &n);

char WordList[n][15];

for (int i = 0; i < n; i++)

{

input(WordList[i]);

}

display(n, WordList);

char word1[15], word2[15];

printf("Enter two words you want to find the distance:\n");

input(word1);

input(word2);

checkpos(n, word1, word2, WordList);

return 0;

}

9. Write a program that asks the user to enter a username. If the username entered is one of the names in the list then the user is allowed to calculate the factorial of a number. Otherwise, an error message is displayed

#include <stdio.h>

#include <string.h>

int check(char user[], char userlist[][15])

{

for (int i = 0; i < 10; i++)

{

if (strcmp(user, userlist[i]) == 0)

return 1;

}

return 0;

}

long long int fac(int n)

{

if (n == 1)

return 1;

else

return n \* fac(n - 1);

}

int main()

{

char userlist[10][15] = {"Rajesh", "Subhajit", "Jhon", "Jake", "Lila", "Riya", "Priya", "SpoilR", "Raja", "Rani"};

printf("Enter your username:\n");

char user[20];

fflush(stdin);

fgets(user, sizeof(user), stdin);

int len = strlen(user);

if (len > 0 && user[len - 1] == '\n')

user[len - 1] = '\0'; // Removing the newline character

int f = check(user, userlist);

if (f)

{

printf("User access granted\nYou can calculate factorial of a number\n");

printf("Enter a number:\n");

int n;

scanf("%d", &n);

printf("%lld", fac(n));

}

else

printf("Access denied\n");

return 0;

}

10. Create an authentication system. It should be menu driven.

#include <stdio.h>

#include <string.h>

#define MAX\_USERS 5

#define MAX\_USERNAME\_LENGTH 20

#define MAX\_PASSWORD\_LENGTH 20

struct User

{

char username[MAX\_USERNAME\_LENGTH];

char password[MAX\_PASSWORD\_LENGTH];

};

struct User userDatabase[MAX\_USERS];

int userCount = 0;

void addUser()

{

if (userCount < MAX\_USERS)

{

printf("Enter new username: ");

scanf("%s", userDatabase[userCount].username);

printf("Enter new password: ");

scanf("%s", userDatabase[userCount].password);

userCount++;

printf("User added successfully.\n");

}

else

{

printf("User database is full. Cannot add more users.\n");

}

}

int authenticate()

{

char username[MAX\_USERNAME\_LENGTH];

char password[MAX\_PASSWORD\_LENGTH];

printf("Enter username: ");

scanf("%s", username);

printf("Enter password: ");

scanf("%s", password);

for (int i = 0; i < userCount; i++)

{

if (strcmp(username, userDatabase[i].username) == 0 &&

strcmp(password, userDatabase[i].password) == 0)

{

return 1; // Authentication successful

}

}

return 0; // Authentication failed

}

int main()

{

int choice;

while (1)

{

printf("Authentication System\n");

printf("1. Add User\n");

printf("2. Authenticate\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

addUser();

break;

case 2:

if (authenticate())

{

printf("Authentication successful.\n");

}

else

{

printf("Authentication failed.\n");

}

break;

case 3:

printf("Exiting program.\n");

return 0;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

}

return 0;

}