Assignment-19 (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>

int main()

{

// char str[8][50] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Extra Day"};

char str[5][50];

int i, j, vowels\_count;

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

{

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

fgets(str[i], 49, stdin);

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

}

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

{ vowels\_count = 0;

for(j = 0; str[i][j]; j++)

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

vowels\_count++;

printf("Number of vowels in %s are %d.\n", str[i], vowels\_count);

}

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>

int main()

{

char str[10][40], new\_str[40];

int i, count = -1;

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

{

printf("Enter name of city %d: ", i + 1);

fgets(str[i], 39, stdin);

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

}

while(count)

{

count = 0;

for(i = 0; i < 9; i++)

if(strcmp(str[i], str[i + 1]) > 0)

{

strcpy(new\_str, str[i]);

strcpy(str[i], str[i + 1]);

strcpy(str[i + 1], new\_str);

count++;

}

}

printf("City names after sorting are:\n");

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

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

printf("\b\b ");

return 0;

}

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

#include<stdio.h>

#include<string.h>

int main()

{

char str[6][40];

int i;

for(i = 0; i < 6; i++)

{

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

fgets(str[i], 39, stdin);

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

}

printf("\nEntered string names are:\n");

for(i = 0; i < 6; i++)

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

printf("\b\b ");

return 0;

}

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

#include<stdio.h>

#include<string.h>

int main()

{

char str[6][40], searchedString[40];

int i;

for(i = 0; i < 6; i++)

{

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

fgets(str[i], 39, stdin);

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

}

printf("Enter the string which is to be searched in the list of strings: ");

fgets(searchedString, 39, stdin);

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

for(i = 0; i < 6; i++)

{

if(strcmp(str[i], searchedString) == 0)

{

printf("String %s found at position %d.", searchedString, i + 1);

break;

}

}

if(i == 6)

printf("String %s not found in list of strings.", searchedString);

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>

int main()

{

char str[6][40];

int i, j;

for(i = 0; i < 6; i++)

{

printf("Enter email-id %d: ", i + 1);

fgets(str[i], 39, stdin);

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

}

for(i = 0; i < 6; i++)

{

for(j = 0; str[i][j]; j++)

{

if(str[i][j] == '@')

break;

}

if(str[i][j] == '\0')

printf("@ missing in %s\n", str[i]);

}

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 reverseString(char str[])

{

int i, str\_length = 0, last\_num;

for(i = 0; str[i]; i++)

str\_length++;

last\_num = str\_length - 1;

for(i = 0; i < str\_length / 2; i++)

{

str[i] = str[i] + str[last\_num];

str[last\_num] = str[i] - str[last\_num];

str[i] = str[i] - str[last\_num];

last\_num--;

}

}

int isStringPalindrome(char s[])

{

char s2[20];

int i;

strcpy(s2, s);

reverseString(s2);

for(i = 0; s[i]; i++)

if(s[i] != s2[i])

return 0;

return 1;

}

int main()

{

char str[6][40];

int i;

for(i = 0; i < 6; i++)

{

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

fgets(str[i], 39, stdin);

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

}

for(i = 0; i < 6; i++)

{

if(isStringPalindrome(str[i]))

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

}

return 0;

}

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

#include<stdio.h>

#include<string.h>

int isIPAddressValid(char ip[])

{

int i, k, dotCount = 0;

char ip1[4], ip2[4], ip3[4], ip4[4];

for(i = 0; ip[i]; i++)

{

if(ip[i] == '.')

dotCount++;

}

if(dotCount == 3)

{

return 1;

}

else

return 0;

}

int main()

{

char str[6][40];

int i;

for(i = 0; i < 6; i++)

{

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

fgets(str[i], 39, stdin);

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

}

for(i = 0; i < 6; i++)

{

if(!isIPAddressValid(str[i]))

printf("%s is not a valid ip address.\n", str[i]);

}

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>

int main()

{

char str[6][40], word1[40], word2[40];

int i, word2ind = -1, k, l, word1ind[6];

for(i = 0; i < 6; i++)

word1ind[i] = -1;

for(i = 0; i < 6; i++)

{

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

fgets(str[i], 39, stdin);

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

}

printf("Enter first word to be considered from the list: ");

fgets(word1, 39, stdin);

word1[strlen(word1) - 1] = '\0';

printf("Enter second word to be considered from the list: ");

fgets(word2, 39, stdin);

word2[strlen(word2) - 1] = '\0';

for(i = 0, k= 0; i < 6; i++)

{

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

{

word1ind[k] = i;

k++;

}

}

l = k - 1;

if(word1ind[l] == 5)

l--;

while(word2ind == -1 && l >= 0)

{

for(i = word1ind[l]; i < 6; i++)

{

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

{

word2ind = i;

printf("Minimum distance between %s and %s is %d.", word1, word2, word2ind - word1ind[l] - 1);

break;

}

}

if(word2ind == -1)

l--;

}

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 factorial(int);

int factorial(int num)

{

int fact = 1;

if(num < 0)

return -1;

else if(num == 0)

return 1;

else

{

for(int i = 1; i <= num; i++)

fact = fact \* i;

return fact;

}

}

int main()

{

char allowedUsers[6][40] = {"User4", "User8", "User12", "User16", "User20", "User24"}, userName[40];

int i, number, fact;

printf("Enter username: ");

fgets(userName, 40, stdin);

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

for(i = 0; i < 6; i++)

{

if(strcmp(allowedUsers[i], userName) == 0)

{

printf("Enter a number to find its factorial: ");

scanf("%d", &number);

fact = factorial(number);

if(fact >= 1)

printf("Factorial of %d is %d.", number, fact);

else

printf("Factorial of %d is undefined.", number);

break;

}

}

if(i == 6)

printf("username error: Please enter a valid user name !\n");

return 0;

}

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

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char users[3][2][40], userPassword[40];

int i, j;

int response;

for(i = 0; i < 3; i++)

{

printf("User %d Registration:\n", i + 1);

printf("Enter username and password for user %d:", i + 1);

for(j = 0; j < 2; j++)

{

fgets(users[i][j], 40, stdin);

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

}

}

while(1)

{

printf("Select one of the usernames to login:\n");

for(i = 0; i < 3; i++)

{

printf("%d. %s\n", i + 1, users[i][0]);

}

printf("0. Exit\n");

scanf("%d", &response);

switch (response)

{

case 1:

printf("Enter password to login: ");

fflush(stdin);

fgets(userPassword, 40, stdin);

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

if(strcmp(users[0][1], userPassword) == 0)

printf("User Authentication Successful !\n");

else

printf("Authentication Failed ! Please Enter correct password next time.\n");

break;

case 2:

printf("Enter password to login: ");

fflush(stdin);

fgets(userPassword, 40, stdin);

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

if(strcmp(users[1][1], userPassword) == 0)

printf("User Authentication Successful !\n");

else

printf("Authentication Failed ! Please Enter correct password next time.\n");

break;

case 3:

printf("Enter password to login: ");

fflush(stdin);

fgets(userPassword, 40, stdin);

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

if(strcmp(users[2][1], userPassword) == 0)

printf("User Authentication Successful !\n");

else

printf("Authentication Failed ! Please Enter correct password next time.\n");

break;

case 0:

exit(0);

default:

printf("Please select correct username by entering the number beside the username !\n");

}

}

return 0;

}