**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>

int main()

{

char str[5][100];

int i, j, vowel=0;

printf("Enter 5 strings:\n");

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

{

fgets(str[i], 100, stdin);

}

printf("\nNumber of vowels in each string:\n");

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

{

for(j=0; str[i][j]!='\0'; 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')

vowel++;

}

printf("%s=%d\n",str[i],vowel);

vowel=0;

}

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][20], temp[20];

int i,j;

printf("Enter 10 cities Name:\n");

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

{

gets(str[i]);

}

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

{

for(j=i+1; j<10; j++)

{

if(strcmp(str[i],str[j])>0)

{

strcpy(temp,str[i]);

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

strcpy(str[j],temp);

}

}

}

printf("Shorted cities Name:\n");

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

{

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

}

return 0;

}

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

#include <stdio.h>

int main() {

char strings[3][10]; // define a 2D array of strings with 3 rows and 10 columns

int i, j;

// read input strings from user

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

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

scanf("%s", strings[i]);

}

// display the 2D array of strings

printf("\nThe array of strings is:\n");

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

for (j = 0; j < 10; j++) {

printf("%c ", strings[i][j]);

}

printf("\n");

}

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[10][50], search\_string[50];

int i, found=0;

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

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

{

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

str[i][strcspn(str[i], "\n")] = '\0'; // Remove newline character

}

printf("Enter the string to search for: ");

fgets(search\_string, sizeof(search\_string), stdin);

search\_string[strcspn(search\_string, "\n")] = '\0'; // Remove newline character

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

{

if(strcmp(str[i], search\_string)==0)

{

printf("String found at index %d\n", i);

found=1;

break;

}

}

if(!found)

{

printf("String not found.\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>

int main() {

char emails[10][30];

int n, i, odd\_index = -1;

printf("Enter the number of email addresses: ");

scanf("%d", &n);

printf("Enter the email addresses:\n");

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

scanf("%s", emails[i]);

if (strchr(emails[i], '@') == NULL) {

printf("Invalid email address: %s\n", emails[i]);

return 0;

}

if (i % 2 == 1 && odd\_index == -1) {

odd\_index = i;

}

}

if (odd\_index != -1) {

printf("Odd email out: %s\n", emails[odd\_index]);

} else {

printf("No odd email out.\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>

int isPalindrome(char \*str);

int main() {

char str[10][50];

int i, n;

printf("Enter the number of strings: ");

scanf("%d", &n);

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

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

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

}

printf("Palindrome strings:\n");

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

if (isPalindrome(str[i])) {

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

}

}

return 0;

}

int isPalindrome(char \*str) {

int len = strlen(str);

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

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

return 0;

}

}

return 1;

}

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

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int isValidIP(char \*ip);

int main() {

char ips[10][16]; // array to store up to 10 IP addresses

int i, n;

printf("Enter the number of IP addresses: ");

scanf("%d", &n);

printf("Enter the IP addresses:\n");

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

scanf("%s", ips[i]);

if (!isValidIP(ips[i])) {

printf("Invalid IP address: %s\n", ips[i]);

return 0;

}

}

printf("All IP addresses are valid.\n");

return 0;

}

int isValidIP(char \*ip) {

char \*a = strtok(ip, ".");

while (a != NULL) {

int x = atoi(a);

if ((x < 0) || (x > 255)) {

return 0;

}

a = strtok(NULL, ".");

}

return 1;

}

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 words[5][20] = {"the", "quick", "brown", "fox", "quick"};

char word1[] = "the";

char word2[] = "fox";

int i, w1 = -1, w2 = -1, min = 10000, temp;

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

{

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

w1 = i;

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

w2 = i;

if(w1 != -1 && w2 != -1)

{

temp = abs(w2 - w1);

if(temp < min)

min = temp;

}

}

printf("%d\n", min);

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>

// List of valid usernames

char usernames[3][20] = {"Aman", "Bhuvan", "Charlie"};

// Function to calculate the factorial of a number

int factorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n-1);

}

}

int main() {

char username[20];

int i, n;

// Prompt the user to enter a username

printf("Enter your username: ");

scanf("%s", username);

// Check if the username is valid

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

if (strcmp(username, usernames[i]) == 0) {

printf("Welcome, %s!\n", username);

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

scanf("%d", &n);

printf("%d! = %d\n", n, factorial(n));

return 0;

}

}

// If the username is not valid, display an error message

printf("Error: Invalid username.\n");

return 0;

}

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

#include <stdio.h>

#include <string.h>

#define MAX\_USERS 10

// Struct to store user information

typedef struct {

char username[20];

char password[20];

} User;

// Function to check if a given username and password match a user in the users array

int authenticate(User users[], int num\_users, char username[], char password[]) {

for (int i = 0; i < num\_users; i++) {

if (strcmp(users[i].username, username) == 0 && strcmp(users[i].password, password) == 0) {

return 1; // Return 1 if username and password match a user in the array

}

}

return 0; // Return 0 if no match is found

}

int main() {

User users[MAX\_USERS]; // Array to store up to MAX\_USERS users

int num\_users = 0; // Counter for number of users currently in array

int choice;

char username[20], password[20];

// Main menu loop

while (1) {

printf("\n1. Add User\n2. Authenticate\n3. Exit\nEnter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

// Add user

if (num\_users == MAX\_USERS) {

printf("Error: Maximum number of users reached.\n");

} else {

printf("Enter username: ");

scanf("%s", users[num\_users].username);

printf("Enter password: ");

scanf("%s", users[num\_users].password);

printf("User added.\n");

num\_users++;

}

break;

case 2:

// Authenticate user

printf("Enter username: ");

scanf("%s", username);

printf("Enter password: ");

scanf("%s", password);

if (authenticate(users, num\_users, username, password)) {

printf("Authentication successful.\n");

} else {

printf("Authentication failed.\n");

}

break;

case 3:

// Exit

printf("Exiting program.\n");

return 0;

default:

printf("Invalid choice.\n");

break;

}

}

}