Q1. 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>

int main()
{
    char arr[5][30];
    int vowel = 0;

    for(int i = 0 ; i < 5; i++)
    {
        printf("\nEnter String %d = ", i+1);
        gets(arr[i]);
    }

    for(int i = 0; i < 5; i++)
    {
        for(int j = 0; arr[i][j] != '\0'; j++)
        {
            if( (arr[i][j] == 'A') || (arr[i][j] == 'a') || (arr[i][j] == 'E') ||(arr[i][j] == 'e') ||(arr[i][j] == 'I')
||(arr[i][j] == 'i') ||(arr[i][j] == 'O') ||(arr[i][j] == 'o') ||(arr[i][j] == 'U') ||(arr[i][j] == 'u') )
            {
                vowel++;
            }
        }

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

        vowel = 0;
    }
}
```

Q2.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 name[10][20];

   char temp[20];


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

   {

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

      gets(name[i]);

   }


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

   {

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

      {

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

         {

            strcpy(temp, name[i]);

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

            strcpy(name[j], temp);

         }

      }

   }


   printf("\n\n");
```

```
    for(int i=0; i<10; i++)
    {
        printf("%d) :- ", i);
        puts(name[i]);
    }
}
```

Q3.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[2][20] = {
                            {"hii, how are you ?"},
                            {"I am fine....."}
                        } ;

    for(int i=0; i<2; i++)
    {
        printf("%s\n",name[i]);
    }
}
```

Q4.Write a program to search a string in the list of strings.

```
#include <stdio.h>
#include <string.h>

int main()
{
    char str[4][20] = {
                            {"hii, how are you ?"},
                            {"I am fine....."},
                            {"What is your name ?"},
                            {"come here please.........!!"}
                        } ;

    char find[20];
    int flag = 0;

    printf("Enter string to search = ");
    gets(find);

    for(int i=0; i<4; i++)
    {
        if(strcmp(str[i], find))
        {
            flag = 0;
        }
        else
        {
            printf("string found");
            flag = 1;
        }
    }


    if(flag == 0)
    {
        printf("string not found");
    }
}
```

Q5. 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[4][40] = {
                            {"jhon45doegmail.com"},
                            {"abc@gmail.com"}
                      };

    char find[20];
    char ch= '@';

    for(int i=0; i<2; i++)
    {
        if(strchr(str[i], '@') != NULL)
        {
            printf("string found");
        }
    }
}
```

Q6. Write a program to print the strings which are palindrome in the list of strings.

```
#include <stdio.h>

#include <string.h>


int main()

{

    char str[5][20] = {

                        {"star"},

                        {"mom"},

                        {"ajay"},

                        {"dad"},

                        {"zerox"}

                    };


    char tmp[20];


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

    {

        strcpy(tmp, str[i]);


        if(strcmp(str[i], strrev(tmp)) == 0)

        {

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

        }

    }

}
```

Q7. From the list of IP addresses, check whether all ip addresses are valid.

```
#include <stdio.h>
#include <string.h>

int main()
{
    char str[5][20] = {
                        {"192.168.1.1"},
                        {"256.7845.1.2"},
                        {"255.106.45.200"},
                        {"1984.150.20.74"},
                        {"4.154.201.0"}
                             };

    int num = 0;
    int flag = 0;

    for(int i = 0; i < 5; i++)
    {
        for(int j = 0; str[i][j] != '\0'; j++)
        {
            if(str[i][0] == '0')
            {
                printf("Wrong IP : %s\n\n", str[i]);
                flag = 1;
                break;
            }
            else if(str[i][j] != '.')
            {
                num = (num * 10) + (str[i][j] - 48);
            }
            else if(num > 255)
            {
                printf("Wrong IP : %s\n\n", str[i]);
                flag = 1;
                num = 0;
                break;
            }
            else
            {
                num = 0;
            }
        }

        if(flag == 0)
        {
            printf("Valid IP : %s\n\n", str[i]);
        }
        else
        {
            flag = 0;
        }

        num = 0;

    }
}
```

Q8.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[5][20] = {

                    {"apple"},

                    {"grapes"},

                    {"house"},

                    {"tiger"},

                    {"grapes"}

                  };


    char word1[20], word2[20];

    int i = 0, count1 = 0, count2 = 0, flag1 = 0, flag2 = 0;


    printf("Enter word1 : ");

    scanf("%s", word1);


    printf("Enter word2 : ");

    scanf("%s", word2);


    for(; i < 5; i++)

    {

        if(!strcmp(str[i], word1) && flag1 == 0)

        {

            count1 = i;
            flag1 = 1;
        }
        else if(!strcmp(str[i], word2) && flag2 == 0)
        {
            count2 = i;
            flag2 = 1;
        }
        else if(flag1 == 1 && flag2 == 1)
        {
            break;
        }
    }

    if(count1<count2)
    {
        printf("Word1 : %d",count1);
    }
    else
    {
        printf("Word2 : %d",count2);
    }
}
```

Q9.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>

void fact(void);

int main()
{
    char str[5][20] = {
                        {"ajay145"},
                        {"rakesgh851"},
                        {"suraj441"},
                        {"ram4110"},
                        {"kishor6321"}
                      };

    char username[20];
    int flag = 1;

    printf("Enter Username : ");
    scanf("%s",username);

    for(int i = 0; i < 5; i++)
    {
        if(!strcmp(str[i], username))
        {
            fact();
            flag = 0;
        }
    }

    if(flag == 1)
    {
        printf("Username not found");
    }
}

void fact(void)
{
    int num = 0, ans = 1;

    printf("Enter number for factorial : ");
    scanf("%d", &num);

    int tmp = num;

    while(tmp)
    {
        ans = (ans * tmp);
        tmp--;
    }
    printf("Factorial of %d : %d", num, ans);

}
```

Q10. Create an authentication system. It should be menu driven.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>

void display(void);
int checkUsername(void);
int checkPassword(void);
void login(void);
void create(void);
void update(void);
void del(void);

int arrayEmpty = 0, flag = 0, chkPass = 0;



char username[5][20];
int ptr = 0;
char password[5][25];

char username_input[50];
char password_input[50];

int main()
{
    while(1)
    {
        display();
    }
}

void display(void)
{
    int choice;

    printf("----------------------Authentication System------------------\n\n");
    printf("\t\t1) Log In\n");
    printf("\t\t2) Create New Username And Password\n");
    printf("\t\t3) Upadate Username And Password\n");
    printf("\t\t4) Delete Username And Password\n");
    printf("\t\t5) Exit\n");
    printf("----------------------Authentication System------------------\n\n");

    printf("Enter Choice -->> ");
    scanf("%d", &choice);

    system("cls");

    switch(choice)
    {
        case 1:
        {
          login();
          break;
        }
        case 2:
        {
          if(ptr < 5)
          {
            create();
```

```
                flag = 1;
            }
            else
            {
                printf("Memory Full");
                getch();
                system("cls");
            }
            flag = 0;
            break;
        }
        case 3:
        {
            update();
            break;
        }
        case 4:
        {
            del();
            break;
        }
        case 5:
        {
            exit(0);
            break;
        }

    }
}

int checkUsername(void)
```

```
{
    if(arrayEmpty == 0)
        return 1;
    else
    {
        for(int i = 0; i < 5; i++)
        {
            if(strcmp(username[i], username_input))
            {
                continue;
            }
            else
            {
                chkPass = i;
                flag = 0;
                return 0;
            }
        }
    }
    return 1;
}

int checkPassword(void)
{
    if(strcmp(password[chkPass], password_input))
    {
        printf("\nWrong Password\n");
        return 0;
    }
    else
    {
        return 1;
    }
}

void login(void)
{
    fflush(stdin);
    printf("Enter Username >>> ");
    gets(username_input);

    if(checkUsername())
    {
        system("cls");
        printf("No Username Exist. Please Create New Username and Password\n");
        getch();
        system("cls");
    }
    else
    {
        printf("Enter Password >>> ");
        gets(password_input);

        if(checkPassword())
        {
            printf("Log In Successfully");
        }
    }
}

void create()
{
    fflush(stdin);

    printf("Enter New Username >>> ");
    gets(username_input);

    if(checkUsername() || flag == 1)
    {
        strcpy(username[ptr], username_input);

        printf("Enter New Password >>> ");
        gets(password[ptr]);

        printf("\nUsername And Password Set Successfully\n");

        ptr = ptr + 1;
        arrayEmpty = 1;
        flag = 0;

        getch();
        system("cls");
    }
    else
    {
        printf("\nUsername And Password Already Set\n");
    }
}

void update(void)
{
    char ans;
```

```
fflush(stdin);
printf("Enter Username >>> ");
gets(username_input);


if(checkUsername())
{
   printf("Username Not Exist");
   getch();
   system("cls");
}
else
{
   printf("If You Want To Update Username ? y/n = ");
   scanf("%c", &ans);

   if(ans == 'y' || ans == 'Y')
   {
      fflush(stdin);
      printf("Enter New Username >>> ");
      gets(username[chkPass]);
   }

   printf("If You Want To Update Password ? y/n = ");
   scanf("%c", &ans);

   if(ans == 'y' || ans == 'Y')
   {
      fflush(stdin);
      printf("Enter New Password >>> ");
      gets(password[chkPass]);
```

```
        }
    }
}

void del(void)
{
    fflush(stdin);
    printf("Enter Username >>> ");
    gets(username_input);

    if(checkUsername())
    {
        printf("Username Not Exist");
        getch();
        system("cls");
    }
    else
    {
        if(chkPass == (ptr-1))
        {
            strcpy(username[chkPass], "NULL");
            strcpy(password[chkPass], "NULL");

            ptr = chkPass;

            printf("Username & Password deleted successfuly");
            getch();
            system("cls");
        }
        else
        {
            strcpy(username[chkPass], username[ptr-1]);
            strcpy(password[chkPass], password[ptr-1]);

            ptr = ptr - 1;

            printf("Username & Password deleted successfuly");
            getch();
            system("cls");
        }
    }
}
```