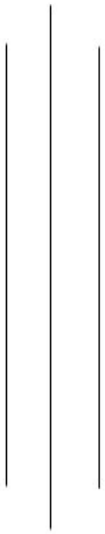
INSTITUTE OF ENGINEERING

Pulchowk Campus, Lalitpur



Subject: C Programming

Lab Report 10

Tittle: **String**

Submitted by:

Susheel Thapa 077BCT090

Submitted to:

Department of Electronics and Computer Engineering

Checked by

Content of Lab Report:

Background Information

C Programming

Editor Used

Compiler

Strings

Code and Output

Source Code

Output

Analysis

Conclusion

Background Information

What is C Programming?

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language.

Why to Learn C Programming?

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Editor

Here, I have used Visual Studio Code as my editor. You can download the editor from Download Visual Studio Code - Mac, Linux, Windows . Select your operating system and download it.

Compiler

Here, I have used **gcc** as my compiler provided by MinGWw64. You can download it via Download MinGW-w64 - for 32 and 64 bit Windows from SourceForge.net. Your download will start automatically. Run the downloaded .exe file. After, you have installed MinGW-w64, you need to configure it.

- In the Windows search bar, type 'settings' to open your Windows Settings.
- Search for Edit environment variables for your account.
- Choose the Path variable and then select Edit.
- Select New and add the Mingw-w64 destination folder path to the system path. The exact path depends on which version of Mingw-w64 you have installed and where you installed it. If you used the settings above to install Mingw-w64, then add this to the path: **C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin.**
- Select OK to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available.

Check your installation

Open command prompt or power shell and type:

```
C:\Users\user>gcc --version
gcc (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
C:\Users\user>gcc
gcc: fatal error: no input files
compilation terminated.

C:\Users\user>_
```

If you get similar result, you are good to go.

String

String is not a fundamental data type in C.
Like many other programming language like javaScript or python in c string is not a fundamental data type. In C string is a sequence of character ending in a \0 character sequence. Thus a string with 10 character is contained in array of 11 character because there must be an \ 0 character at the end for an string to exist.

If there are other characters after the \0 character those character are ignored and only other character are displayed.

\0 character indicates the end of string.

char a[10]={‘r’,’a’,\0,’m’};

Here a is a string with only 2 characters and m is not part of the string although it is part of a the array

There are a lot of string handling function must of these function accepts the first argument as a pointer which represent the first element of array.

Strlen():

Syntax:
strlen(stringVariable)
is a string handaling function which accepts the character pointer as its first element and returns the length of the string as output.

The important property about strlen() is that it looks for the first \0 character and does not proceed further.

strlen() function is defined in <string.h> header file.
int a= strlen(a);

strcat():
the function prototype for strcat() is char* strcat(char* dest,char * src)
This means that the function appends the string in the src at the end of the dest and returns the dest pointer too.
The important precaution to betaken is src should be large enough to include the both first and second string.
The function adds \0 character at the end of resut but if the dest is small \0 might not be included and it might endup being just array of characters.

strcpy()
strcpy() is a function defined in <string.h> to copy a string from one destination to another we can use the strcpy() function for this purpose,
Syntax
char *strcpy(char *dest, const char *src)
The function copies the string of the src to the string dest and returns the pointer to dest.

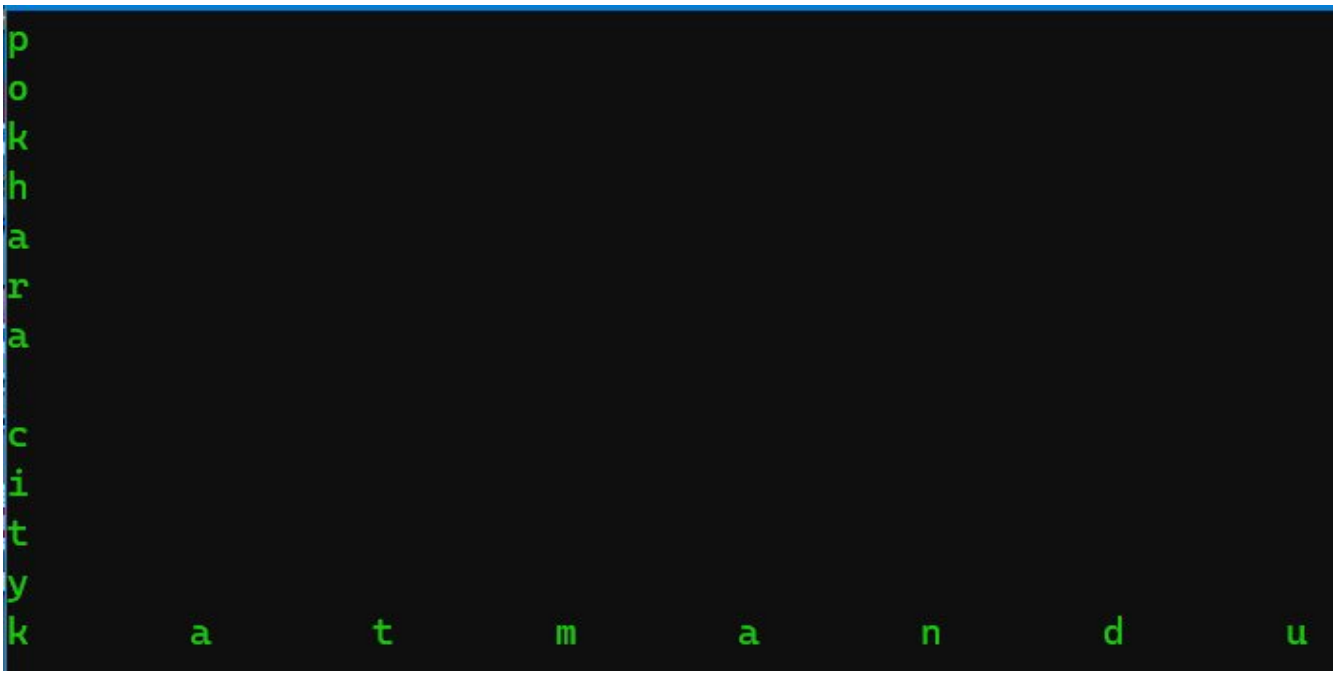
strcmp()
strcmp() function takes two function as argument and returns the
int strcmp(const char *str1, const char *str2)
the function is defined in <string.h> and is used to analyze the two string it returns 0 if str1 and str2 are completely identical
it returns a negative number if str2 is less than str2 and a positive number if str1 is greater than str2.

1.Write the following program , observe the output and comment on it
Source Code

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

void main()
{
   /*Variable Declaration and Initialization*/
   int i;
```

```
    char name1[] = "pokhara city";
    char name2[] = {'k', 'a', 't', 'm', 'a', 'n', 'd', 'u', '\0'};

    system("cls");

    /*Prints the name 1 in vertical manner*/
    for (i = 0; i < strlen(name1); i++)
    {
        printf("%c\n", name1[i]);
    }

    /*Prints the name 2 in horizontal way with a gap of one tab*/
    for (i = 0; i < strlen(name2); i++)
    {
        printf("%c\t", name2[i]);
    }

    getch();
}
```

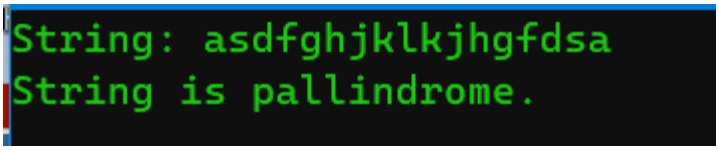
Output:



2.Write a program whether the string enter by the user is Pallindrome or not
Source Code

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

int main()
{
   /*Variable Declaration*/
   char string[100], rev_string[100];

   system("cls");

   /*Taking string*/
   printf("String: ");
   scanf("%[^'\n']", string);

   /*Copy string to new variable and reversing it*/
   strcpy(rev_string, string);
   strrev(rev_string);

   /*Condtion of pallindrome*/
   if (strcmp(rev_string, string) == 0)
   {
      printf("String is pallindrome.\n");
   }
   else
   {
      printf("String isn't pallindrome.\n");
   }

   getch();
   return 0;
}
```

Output:



3.Write a program to read a string in main, pass it to a function that returns the count of numbers of words to main(). Display the count
Source Code

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

int wordCount(char string[])
{
   "Count the words in string and return the counted words";
   int words = 0;

   for (int i = 0; i < strlen(string); i++)
   {
      if (isalpha(string[i]) && (string[i + 1] == ' ' || string[i + 1] == '\t' || string[i + 1] == '\
0'))
      {
         words++;
      }
   }

   return words;
}
int main()
{
   char sentence[100];

   system("cls");

   /*Taking string*/
```

```
    printf("String: ");
    scanf("%[^\n']", sentence);

    /*Printing the result and calling function*/
    printf("Number of words is %d.", wordCount(sentence));

    getch();
    return 0;
}
```

Output:

```
String: i am god
Number of words is 3.
```

4.TO find the highest word in the string
Source Code

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

void calculate(char sentence[10000], int *vowel, int *consonant, int *length, char
longest[100])
{
    char temp[100][20];
    int position = 0;
    int i = 0, j = 0, k = 0;

    int v = 0, c = 0, l = 0;

    while (1)
    {
    NextIteration:

        /*Count the length of words, vowel, consonant*/
        if (isalpha(sentence[j]))
```

```
    {
        if (sentence[j] == 'a' || sentence[j] == 'e' || sentence[j] == 'i' || sentence[j] == 'o' ||
sentence[j] == 'u' || sentence[j] == 'A' || sentence[j] == 'E' || sentence[j] == 'I' || sentence[j]
== 'O' || sentence[j] == 'U')
        {
            v++;
            l++;
            temp[i][k] = sentence[j];
        }

        else if (sentence[j] != 'a' || sentence[j] != 'e' || sentence[j] != 'i' || sentence[j] != 'o' ||
sentence[j] != 'u' || sentence[j] != 'A' || sentence[j] != 'E' || sentence[j] != 'I' || sentence[j] !=
'O' || sentence[j] != 'U')
        {
            c++;
            l++;
            temp[i][k] = sentence[j];
        }
    }

    if (sentence[j] == ' ' || sentence[j] == '.' || sentence[j] == '\0')
    {
        temp[i][k] = '\0';

        if (i != 0)
        {
            if (strlen(temp[position]) < strlen(temp[i]))
            {
                position = i;

                *vowel = v;
                *consonant = c;
                *length = l;
            }
        }
        if (i == 0)
        {
            *vowel = v;
            *consonant = c;
            *length = l;
        }
```

```
            v = 0;
            c = 0;
            l = 0;

            i++;
            k = 0;
            j++;

            if (sentence[j] == '\0')
            {
               break;
            }

            goto NextIteration;
         }

         if (sentence[j] == '\0')
         {
            break;
         }

         j++;
         k++;
      }

      strcpy(longest, temp[position]);
}
int main()
{
      char string[200], longest[200];
      int vowel, consonant, length;

      system("cls");
      printf("String: ");
      gets(string);

      calculate(string, &vowel, &consonant, &length, longest);

      printf("\nVowel =%d,\nConsonant = %d, \nLength = %d \nWords = %s\n", vowel,
consonant, length, longest);

      getch();
```

```
    return 0;
}
```

Output:

```
Sentence: My name is susheel thapa
Longest Word: susheel
Vowel Count: 3
Consonents Count: 7
```

5.Write a program to reverse the word using recursive function
Source Code

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

void reverse(char string[], int length)
{
    "Reverse the string recursively based on length of string is even or odd";

    /*Find the length of string and it is constant so it is stored in const integer variable*/
    const int length_of_string = strlen(string);

    /*Check length is odd or even as when have seperate process of reversing odd and even length of string*/
    if (strlen(string) % 2 == 0)
    {

        /*Recursive termination condition*/
        if (length == length_of_string / 2)
        {
            return;
        }
        else
        {
            /*Swap the element of first and last position*/
            char temp = string[length_of_string - length];
            string[length_of_string - length] = string[length - 1];
            string[length - 1] = temp;
```

```
            /*Decrease length by 1*/
            length = length - 1;

            /*Function call*/
            reverse(string, length);
        }
    }
    else if (length_of_string % 2 == 1)
    {
        /*Finding midpoint as in odd length midpoint element isn't changed*/
        const int midpoint = (length_of_string + 1) / 2;

        /*Terminating condition*/
        if (length == midpoint)
        {
            // goto stop;
            return;
        }
        else
        {
            /*Swap the last and first element*/
            char temp = string[length_of_string - length];
            string[length_of_string - length] = string[length - 1];
            string[length - 1] = temp;

            /*Decrese length by 1*/
            length = length - 1;

            /*Function call*/
            reverse(string, length);
        }
    }
}

int main()
{
    /*Variable declaration*/
    char string[100];

    system("cls");
```

```
    /*Taking string from user*/
    printf("String: ");
    scanf("%[^'\n']", string);

    /*Function call*/
    reverse(string, strlen(string));

    /*Reversing the string*/
    printf("Reversed string is %s\n", string);

    getch();
    return 0;
}
```

Output:

```
String: Susheel Thapa
Reversed string is apahT leehsuS
```

6.Write a seperate program that exactly stimulate the task of strlen(),strcat(),strcpy() and strcmp() using \nuser defined function
Source Code

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

/*All these function are avaliable in string folder*/
void combine(char *str_one, char *str_two)
{
    "Combine the two string and kept combine result in first variable";

    while (*str_one != '\0')
    {
        str_one++;
    }
```

```
    while (*str_two != '\0')
    {
        *str_one = *str_two;
        str_one++;
        str_two++;
    }
    *str_one = '\0';
}

int compareString(char *string_one, char *string_two)
{
    while (1)
    {
        if (*string_one != *string_two)
        {
            return 1;
        }
        else if (*string_one == *string_two)
        {
            if (*string_one == '\0' || *string_two == '\0')
            {
                return 0;
            }
        }
        if (*string_one == '\0' || *string_two == '\0')
        {
            return 1;
        }

        string_one++;
        string_two++;
    }
}

void copyString(char *destination, char *source)
{
    "Copy string from one variable to another";

    while (*source != '\0')
    {
        *destination = *source;
        destination++;
```

```
        source++;
    }

    *destination = '\0';
}

int lengthString(char str[])
{
    int length = 0;
    while (str[length] != '\0')
    {
        length++;
    }
    return length;
}

int main()
{
    /*Vaeiable Declaration*/
    char string1[100], string2[100], temp[100];

    system("cls");

    /*Taking two string from user*/
    printf("String one: ");
    scanf("%[^'\n']", string1);

    fflush(stdin); /*To clear buffer*/

    printf("String two: ");
    scanf("%[^'\n']", string2);

    /*Copying value to temp as string2 value will be change later on*/
    copyString(temp, string2);

    /*Calling each function and printing meaningful result*/
    printf("\nUser Defined Length\n");
    printf("Length of string one is %d\n", lengthString(string1));
    printf("Length of string two is %d\n", lengthString(string2));

    printf("\nUser Defined Compare\n");
    if (compareString(string1, string2) == 0)
```

```
    {
        printf("Two string is same.\n");
    }
    else
    {
        printf("Two string isn't same.\n");
    }

    printf("\nUser Defined Copy\n");
    copyString(string2, string1);
    printf("Copying items of string one to string two: %s\n", string2);

    /*Since string is change so we reinitialized it*/
    copyString(string2, temp);

    printf("\nUser Defined Combine\n");
    combine(string1, string2);
    printf("Combination of string one and string two is %s\n", string1);

    getch();
    return 0;
}
```

Output:

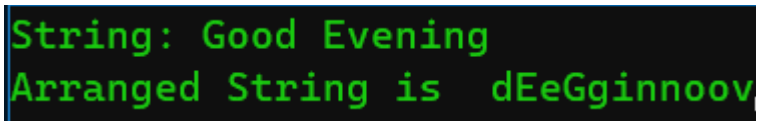
```
String one: Good Evening
String two: Good Night

User Defined Length
Length of string one is 12
Length of string two is 10

User Defined Compare
Two string isn't same.

User Defined Copy
Copying items of string one to string two: Good Evening

User Defined Combine
Combination of string one and string two is Good EveningGood Night
```

7.Write a program that reads string and arranged it in alphabetical order
Source Code

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

void arrangeAlphabetical(char string[])
{
   "Arrange the string in alphabetical order. Use bubble sort algorithm";

   for (int i = 0; i < strlen(string); i++)
   {
      for (int j = 0; j < strlen(string); j++)
      {
         /*Below if will convert first character to lower and then compare so the ascending
order is arranged*/
         if (((int)tolower(string[i]) - (int)tolower(string[j])) < 0)
         {
            char temp = string[i];
            string[i] = string[j];
            string[j] = temp;
         }
      }
   }
}

int main()
{
   /*Varaible Declaration*/
   char string[100];

   system("cls");

   /*Taking input*/
   printf("String: ");
   scanf("%[^'\n']", string);

   /*Arranging String*/
   arrangeAlphabetical(string);
```

```
    /*Printing result*/
    printf("Arranged String is %s", string);

    getch();
    return 0;
}
```

Output:

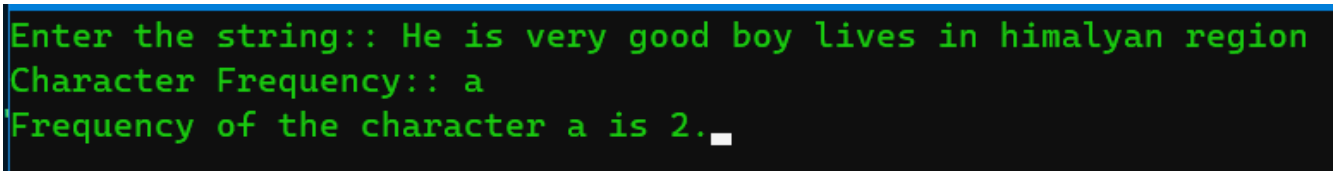


8.Write a program to find the frequency of the character in the string enter by the user
Source Code

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

int main()
{
    char string[100];
    char ch;
    int i = 0, count = 0;

    system("cls");

    printf("Enter the string:: ");
    gets(string);

    printf("Character Frequency:: ");
    scanf("%c", &ch);

    while (string[i] != '\0')
    {
        if (string[i] == ch)
        {
            count++;
        }
```

```
        i++;
    }

    printf("Frequency of the character %c is %d.", ch, count);

    getch();
    return 0;
}
```

Output

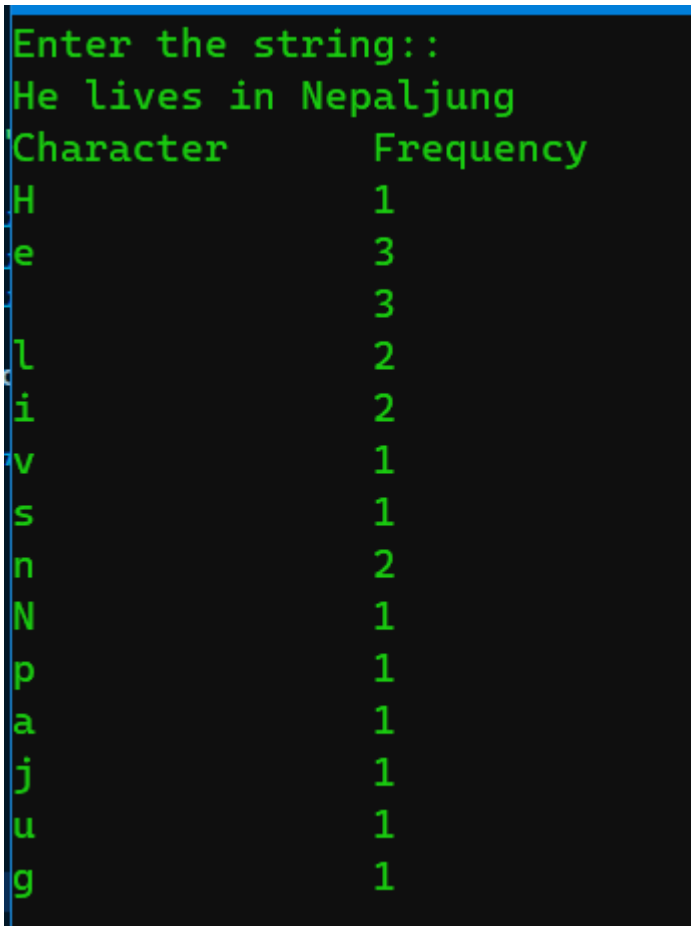


9.Write a program to find the frequency of the characters in the string enter by the user
Source Code

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

struct Frequency
{
    char alphabet;
    int count;

} Character[26];

int main()
{
    char string[100];
    int i = 0, number_of_character = 0;

    system("cls");

    printf("Enter the string::\n");
    gets(string);
```

```
    while (string[i] != '\0')
    {
        if (i == 0)
        {
            Character[0].alphabet = string[i];
            Character[0].count = 1;
            number_of_character++;
            goto Last;
        }

        for (int j = 0; j < number_of_character; j++)
        {
            if (string[i] == Character[j].alphabet)
            {
                Character[j].count++;
                goto Last;
            }
        }

        Character[number_of_character].alphabet = string[i];
        Character[number_of_character].count = 1;
        number_of_character++;
    Last:
        i++;
    }

    printf("%-15s%-15s\n", "Character", "Frequency");

    for (i = 0; i < number_of_character; i++)
    {
        printf("%-15c%-15d\n", Character[i].alphabet, Character[i].count);
    }

    getch();
    return 0;
}
```

Output:

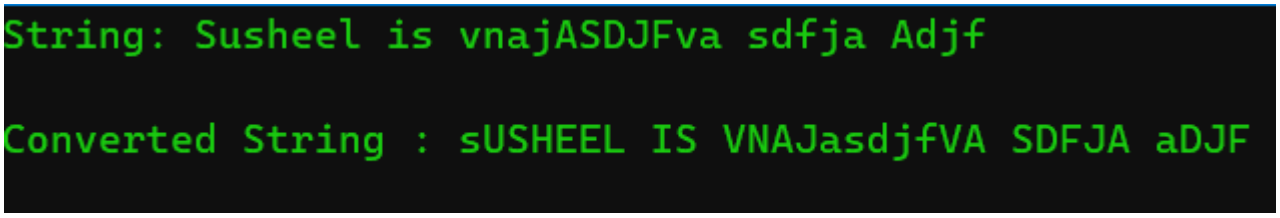


10.Write a program to read a string in main(). Pass it to function. The fucntion convert all the uppercase of lower and vice-verse
Source Code

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

void convertLowerUpper(char *string)
{
   "Convert the lower case to uppercase and vice verse";

   while (*string != '\0')
   {
```

```
        (isupper(*string)) ? (*string = tolower(*string)) : (*string = toupper(*string));
        string++;
    }
}

int main()
{
    /*Variable Declaration*/
    char string[100];

    system("cls");

    /*Taking string input*/
    printf("String: ");
    scanf("%[^'\n]", string);

    /*Function call*/
    convertLowerUpper(string);

    /*Printing result*/
    printf("\nConverted String : %s\n", string);

    getch();
    return 0;
}
```

Output:

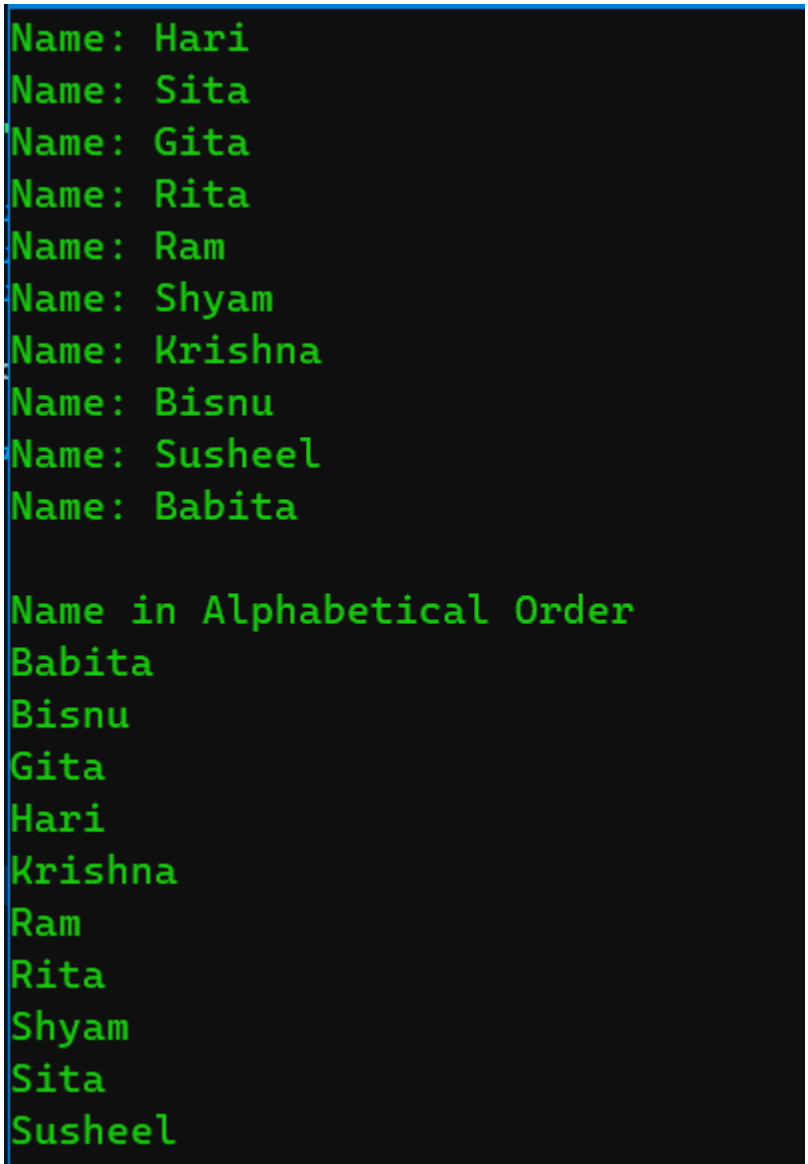


11.Write a program to read the name of 10 students in main(). Pass the name list to function that sorts the array to ascending order. Display the sorted array from main().
Source Code

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

void sort(char name[][100])
{
   "Compare each string and arrange in alphabatical order. BUBBLE SORT TECHNIQUE";

   char temp[100];
   for (int i = 0; i < 10; i++)
   {
      for (int j = 0; j < 10; j++)
      {
         if (strcmpi(name[i], name[j]) < 0)
         {
            strcpy(temp, name[i]);
            strcpy(name[i], name[j]);
            strcpy(name[j], temp);
         }
      }
   }
}

int main()
{
   /*Variable Declaration*/
   char student_name[10][100];

   system("cls");

   /*Taking Input*/
   for (int i = 0; i < 10; i++)
   {
      fflush(stdin);

      printf("Name: ");
      scanf("%[^'\n']", student_name[i]);
```

```
    }

    /*Function call*/
    sort(student_name);

    /*Printing Result*/
    printf("\nName in Alphabetical Order\n");
    for (int i = 0; i < 10; i++)
    {
        printf("%s\n", student_name[i]);
    }

    getch();
    return 0;
}
```

Output:



12
Write a program to do following things:
-->To print the question "Who is prime minister of Nepal?"
-->To accept answer
-->TO print "Good" and stop if the answer is correct
-->To print the message "try again", if the answer is wrong.
-->To display the correct answer when the answer is wrong even at the third attempt, exit the program
Source Code

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

int main()
{
   /*Variable Declaration*/
   char answer[50];
   int attempt = 0;

   system("cls");

   /*While loop to take,check answer and print infomation on the answer and stop when attempt is 3*/
   while (attempt != 3)
   {
      fflush(stdin); /*CLear Buffer*/

      /*Asking Question and receivng answer*/
      printf("Who is prime minister of Nepal?\n");
      scanf("%[^'\n']", answer);

      /*Checking Answer and taking decision*/
      if (strcmpi(answer, "Pushpa Kamal Dahal") == 0)
      {
         printf("Good");
         break;
      }
      else
      {
         system("cls");
```

```
            printf("Try again\n");
            attempt++;
        }
    }

    /*If user gives all wrong answer*/
    if (attempt == 3)
    {
        printf("\nYou have tried for 3 times but you were unable to give correct answer.\n");
        printf("\nCorrect Answer: %s", "'Pushpa Kamal Dahal'");
    }

    getch();
    return 0;
}
```

Output: