Exercise 1 – *Write a program which will count number of vowels, consonants and digits from any given string. An example would be as follows:*

```
Enter a string: My ph. No. is 1234567890.

Number of vowels: 2
Number of consonants: 6
Number of digits: 10
```

Program –

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];
    int vowels=0,consonants=0,digits=0,i,length;
    printf("Enter a string: ");
    scanf("%[^\n]%*c", &str);
    length=strlen(str);
    for(i=0;i<length;i++)
    {
        if(str[i]=='a' || str[i]=='A' || str[i]=='e' || str[i]=='E' || str[i]=='i' || str[i]=='I' || str[i]=='o' ||
str[i]=='O' || str[i]=='u' || str[i]=='U')
        {
            vowels++;
        }
        else if((str[i]>='A' && str[i]<='Z') || (str[i]>='a' && str[i]<='z'))
        {

            consonants++;
        }
        else if(str[i]>='0' && str[i]<='9')
        {
            digits++;
        }
        else
```

```
        {
            continue;
        }
    }
    printf("\nNumber of vowels: %d\n", vowels);
    printf("Number of consonants: %d\n", consonants);
    printf("Number of digits: %d\n", digits);
    return 0;
}
```

Output –

```
Enter a string: My ph. No. is 1234567890.

Number of vowels: 2
Number of consonants: 6
Number of digits: 10
```

Exercise 2 – *Write a program to check whether a given string is palindrome or not. An example would be as follows:*

```
        Enter a word: Madam
        Given word is palindrome.

        Enter a word: Hello
        Given word is not palindrome.
```

Program –

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main()
{
    char str[100];
    int flag=0,i,length;
```

```
        printf("Enter a word: ");
        scanf("%s", &str);
        length=strlen(str);
        for(i=0;i<length;i++)
        {
                if((str[i] == str[length-i-1]) || (str[i] == toupper(str[length-i-1])) || (str[i] == tolower(str[length-i-
1])))

                {
                        continue;
                }
                else
                {
                        flag = 1;
                        break;
                }
        }
        if(flag == 1)
        {
                printf("Given word is not palindrome.");
        }
        else
        {
                printf("Given word is palindrome.");
        }
        return 0;
    }
```

Output –

```
    Enter a word: Madam
    Given word is palindrome.

    Enter a word: Hello
    Given word is not palindrome.
```

Exercise 3– *Write a program to abbreviate name. An example would be as follows:*

```
            Enter the full name: Rabindra Nath Tagore
            R.N.Tagore
```

Program –

```c
#include <stdio.h>
#include <string.h>
int main()
{
      char str[100];
      int vowels=0,consonants=0,digits=0,i,length,a;
      printf("Enter the full name: ");
      scanf("%[^\n]%*c", &str);
      printf("%c.",str[0]);   //First Name
      length=strlen(str);
      for(i=0;i<length;i++)   //To find last null place
      {
            if(str[i] == ' ')
            {
                  a=i;
            }
      }
      for(i=0;i<a;i++)        //Middle Name
      {
            if(str[i] == ' ')
            {
                  printf("%c.",str[i+1]);
            }
      }
      for(i=a+1;i<length;i++) //Last Name
      {
            printf("%c",str[i]);
      }
      return 0;
}
```

Output –

```
Enter the full name: Rabindra Nath Tagore
R.N.Tagore
```

Exercise 4– *Write the following user defined function to perform the corresponding job.*

Function	Job
xStrlen():	Find the length of a string.
xStrcpy():	Copy a string from a source to target.
xStrcmp():	Compare two strings whether they are identical or not.
xStrcat():	Merge two strings.

Solution –

Function	Function Definition	Example
xStrlen():	```int xStrlen(char *str1)
{
int length=0,i;
for(i=0;str1[i]!='\0';i++)
{
length++;
}
return length;
}```	```#include <stdio.h>
int xStrlen(char *str1)
{
int length=0,i;
for(i=0;str1[i]!='\0';i++)
{
length++;
}
return length;
}
void main()
{
char str1[100];
int length = 0;
printf("Enter a string: ");
scanf("%[^\n]%*c", str1);```

|  |  | ```
length=xStrlen(str1);
printf("\nLength = %d\n", length);
}
``` |
| --- | --- | --- |
| **xStrcpy():** | ```
char xStrcpy(char *str2, char
*str1)
{
    int i;
    for(i=0;str1[i]!='\0';i++)
    {
        str2[i] = str1[i];
    }
    str2[i] = '\0';
}
``` | ```
#include <stdio.h>
char xStrcpy(char *str2, char *str1)
{
    int i;
    for(i=0;str1[i]!='\0';i++)
    {
        str2[i] = str1[i];
    }
    str2[i] = '\0';
}
void main()
{
    char str1[100],str2[100];
    printf("Enter a string1: ");
    scanf("%[^\n]%*c", str1);
    xStrcpy(str2,str1);
    printf("\nString 1: %s\nString 2: %s\n",str1,
str2);
}
``` |
| **xStrcmp():** | ```
int xStrcmp(char *str1, char *str2)
{
    int flag=0,i;
    for(i=0;str1[i]!='\0';i++)
    {
        if(str1[i] != str2[i])
        {
            flag = 1;
            break;
        }
    }
``` | ```
#include<stdio.h>
int xStrcmp(char *str1, char *str2)
{
    int flag=0,i;
    for(i=0;str1[i]!='\0';i++)
    {
        if(str1[i] != str2[i])
        {
            flag = 1;
            break;
``` |

```
        return flag;
}
```

```
        }
    }
    return flag;
}
int main()
{
    char str1[100],str2[100];
    int flag;
    printf("Enter a string1: ");
    scanf("%[^\n]%*c", str1);
    printf("Enter a string2: ");
    scanf("%[^\n]%*c", str2);
    flag=xStrcmp(str1,str2);
    if(flag == 1)
    {
        printf("\nMismatched.\n");
    }
    else
    {
        printf("\nMatched.\n");
    }
    return 0;
}
```

xStrcat():	```char *xStrcat(char *str1,char *str2)
{
int i,j;
for(i=0;str1[i]!='\0';i++)
{
}
for(j=0;str2[j]!='\0';j++)
{
str1[i++] = str2[j];```	```#include <stdio.h>
char *xStrcat(char *str1,char *str2)
{
int i,j;
for(i=0;str1[i]!='\0';i++)
{
}
for(j=0;str2[j]!='\0';j++)
{```

```
        }
        str1[i] = '\0';
        return str1;
}
```

```
            str1[i++] = str2[j];
        }
        str1[i] = '\0';
        return str1;
}
void main()
{
    char str1[100],str2[100];
    printf("Enter a string1: ");
    scanf("%[^\n]%*c", str1);
    printf("Enter a string2: ");
    scanf("%[^\n]%*c", str2);
    xStrcat(str1,str2);
    printf("\n%s\n",str1);
}
```