



**NAME: RUCHIRA NASKAR**

**ROLL NO.: 21075072**

**DEPARTMENT: COMPUTER SCIENCE AND  
ENGINEERING**

**SECTION: BA1**

**CSO 101 LAB ASSIGNMENT-7:**

**RECURSIVE FUNCTIONS**

**EMAIL ID: [ruchira.naskar.cse21@iitbhu.ac.in](mailto:ruchira.naskar.cse21@iitbhu.ac.in)**

## SOLUTIONS

1) Use recursive function to find the length of a string.

CODE:

```
#include<stdio.h>
int lstr(char x[],int i)
{
    if (x[i]=='\0')
        return 0;
    else
        return 1+lstr(x,i+1);
}
int main()
{
    char str[1000];
    scanf("%[^\n]s", str);
    fflush(stdin);
    int v=lstr(str,0);
    printf("The length of the string is %d.",v);
    return 0;
}
```

OUTPUT:

```
hELLO
The length of the string is 5.
PS C:\Users\91933\Documents\CSO\Assignments\Lab assignments\S7> █
```

2) Compare two strings using recursive functions and print which occurs alphabetically first.

CODE:

```
#include<stdio.h>
int comp(char x[],char y[],int i)
{
    if (x!='\0' && y!='\0' && x[i]==y[i])
        return comp(x,y,i+1);
    else if ((x!='\0' && y=='\0') || ((x!='\0' && y!='\0') &&
x[i]<y[i]))
        return 1;
    else if ((x=='\0' && y!='\0') || ((x!='\0' && y!='\0') &&
x[i]>y[i]))
        return 2;
    else if (x=='\0' && y=='\0')
        return 0;
}
int main()
{
    char a[1000],b[1000];
    printf("The first string a is: ");
    scanf("%[^\n]s",&a);
    fflush(stdin);
    printf("The second string b is: ");
    scanf("%[^\n]s",&b);
    fflush(stdin);
    int v=comp(a,b,0);
    if (v==0)
        printf("The strings are equal.\n");
    else if (v==1)
        printf("String %s comes first.\n",a);
    else if (v==2)
        printf("String %s comes first.\n",b);
    return 0;
}
```

OUTPUT:

```
.\42
The first string a is: heya
The second string b is: abc
String abc comes first.
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> █
```

3) Use recursive function to find product of digits of a number.

CODE:

```
#include<stdio.h>
int prod(int n)
{
    if (n==0)
        return 1;
    else
        return (n%10)*prod(n/10);
}
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    fflush(stdin);
    int p;
    p=prod(n);
    printf("The product of digits of the number is: %d",p);
    return 0;
}
```

OUTPUT:

```
Enter a number: 2341
The product of digits of the number is: 24
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> 
```

4) Use recursive function to find if a number is prime or not.

CODE:

```
#include<stdio.h>
int prime(int n,int i)
{
    if (n<3)
        return (n==2)?1:0;
    else if (i==n)
        return 1;
    else if (n%i==0)
        return 0;
    else
        return prime(n,i+1);
}
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    fflush(stdin);
    int flag=prime(n,2);
    if (flag==1)
        printf("The number %d is prime.\n",n);
    else
        printf("The number %d is composite.\n",n);
    return 0;
}
```

OUTPUT:

```
PS C:\Users\91933\Documents\CSO\Assignments\Lab assignments\S7> cd "c:\Us
.\q4 }
Enter a number: 23
The number 23 is prime.
PS C:\Users\91933\Documents\CSO\Assignments\Lab assignments\S7> cd "c:\Us
.\q4 }
Enter a number: 32
The number 32 is composite.
PS C:\Users\91933\Documents\CSO\Assignments\Lab assignments\S7> █
```

5) Find the factors of a number using recursive functions.

CODE:

```
#include<stdio.h>
int factors(int n,int i)
{
    if (i==n+1)
        return 0;
    else
    {
        if (n%i==0)
            printf("%d ",i);
            factors(n,i+1);
        }
    }
}
int main()
{
    int n;
    scanf("%d",&n);
    fflush(stdin);
    printf("The factors of the number %d are:-\n",n);
    factors(n,1);
    return 0;
}
```

OUTPUT:

```
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> cd ".\q5"
50
The factors of the number 50 are:-
1 2 5 10 25 50
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> cd ".\q5"
31
The factors of the number 31 are:-
1 31
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> 
```

6) Find the GCD of two numbers using recursive functions.

CODE:

```
#include<stdio.h>
int gcd(int p,int q)
{
    if (q==0)
        return p;
    return gcd(q,p%q);
}
int main()
{
    int p,q;
    printf("Enter the numbers: ");
    scanf("%d %d",&p,&q);
    int g;
    int x,y;
    x=p,y=q;
    if (p<q)
        x=q,y=p;
    g=gcd(x,y);
    printf("The GCD of the numbers is: %d",g);
    return 0;
}
```

OUTPUT:

```
Enter the numbers: 25 20
The GCD of the numbers is: 5
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7>
```

7) Check if a string is palindrome or not using recursive functions.

CODE:

```
#include<string.h>
#include<stdio.h>
int pal(char x[],int i, int f)
{
    if (i==f)
        return 1;
    else if (x[i]==x[f])
        return pal(x,i+1,f-1);
    else if (x[i]!=x[f])
        return 0;
}
int main()
{
    char a[1000];
    printf("Enter the string: ");
    scanf("%[^\n]s",&a);
    fflush(stdin);
    int v=pal(a,0,strlen(a)-1);
    if (v==0)
        printf("The string is not palindrome.\n");
    else if (v==1)
        printf("The string is palindrome.\n");
    return 0;
}
```

OUTPUT:

```
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> cd "c:\U
.\q7 }
Enter the string: mom
The string is palindrome.
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> cd "c:\U
.\q7 }
Enter the string: harry
The string is not palindrome.
PS C:\Users\91933\Documents\CS0\Assignments\Lab assignments\S7> █
```