

# Programming Fundamentals



Department of Computer Engineering

Khwaja Fareed University of Engineering and  
Information Technology  
Rahim Yar Khan, Pakistan

---

# Contents

---

1	Week 1&2: The Programming Environment	1
2	Week 3&4: Basic Decision Constructs and for loop	7
3	Week 5: while Loops & Sentinel Controlled Repetition	12
4	Week 6&7: Functions	17
5	Week 8: Manipulating Arrays	23
6	Week 9: Pointers Notations and Arrays	26
7	Week 10: Pointers	29
8	Week 11: Strings	32
9	Week 12: Structures	34
10	Week 13: Structures	36
11	Week 14&15: File Management	38

---

# Week 1&2: The Programming Environment

---

## Learning Objectives:

The objectives of this first experiment are:

1. To make you familiar with the Windows programming environment and the "C" compiler. For this purpose you will use DevCPP/Code Blocks. You will write, compile and run a (it may be your first) program in "C"
2. To get you to write a number of simple algorithms using flowcharts

## Exercise 1:

Print following shape using simple printf statements

(1)	(2)	(3)	(4)	(5)	(6)
*	*****	*	*	*****	*****
***	*       *	**	**	****	****
*****	*       *	***	***	***	***
***	*       *	****	****	**	**
*	*****	*****	*****	*	*

Figure 1.1: Printing Triangle

## Code:

(1)

```
#include <stdio.h>
int main ()
{
    printf (" *");
```

```
    printf ("\n ***");
    printf ("\n*****");
    printf ("\n ***");
    printf ("\n  *");
    return 0;
}
```

(2)

```
#include <stdio.h>
int main ()
{
    printf ("*****");
    printf ("\n*\t*");
    printf ("\n*\t*");
    printf ("\n*\t*");
    printf ("\n*\t*");
    printf ("\n*****");
    return 0;
}
```

(3)

```
#include <stdio.h>
int main ()
{
    printf (*);
    printf ("\n**");
    printf ("\n***");
    printf ("\n****");
    printf ("\n*****");
    return 0;
}
```

(4)

```
#include <stdio.h>
int main ()
{
    printf ("  *");
    printf ("\n  **");
    printf ("\n   ***");
    printf ("\n  ****");
    printf ("\n*****");
    return 0;
}
```

(5)

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

---

```
int main ()
{
    printf ("*****");
    printf ("\n*****");
    printf ("\n***");
    printf ("\n**");
    printf ("\n*");
    return 0;
}
```

**(6)**

```
#include <stdio.h>
int main ()
{
    printf ("*****");
    printf ("\n *****");
    printf ("\n  ***");
    printf ("\n   **");
    printf ("\n    *");
    return 0;
}
```

## Exercise 2:

Write a C-Program to perform the simple arithmetic operations (addition, subtraction, multiplication, division, remainder).

**Code:**



### Exercise 3:

Write a C-Program to swap two integer numbers without and with using third variable.

#### Code:

```
#include<stdio.h>
int main()
{
    int x=5,y=10;
    int hold;
    printf("\nBefore swapping x = %d, y = %d",x,y);
    x = x + y;
    y = x - y;
    x = x - y;
    printf("\nAfter swapping without using third variable x = %d, y = %d", x ,y);

    printf("\nBefore swapping x = %d, y = %d",x,y);
    hold = x;
    x = y;
    y = hold;
    printf("\nAfter swapping with using third variable x = %d, y = %d", x ,y);
    return 0;
}
```

### Exercise 4:

Write a C-Program to calculate area and Perimeter of the triangle and rectangle. [Area of triangle=  $1/2 \times \text{base} \times \text{vertical height}$  , Perimeter of triangle =  $a + b + c$ ]  
[Area of rectangle=  $\text{width} \times \text{height}$  , Perimeter of rectangle =  $2(a + b)$ ]

#### Code:

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    printf ("-----For Triangle-----\n");
    float at,pt,b,vh,s1,s2; /*where at=Area of Triangle, pt=Perimeter Of Triangle
    b=Base, vh=Verticle Height,
    s1=Side1, s2=Side2,*/
    b=18;
    vh=12;
    s1=15;
    s2=15;
    at= 0.5*b*vh;
    pt= s1+s2+b; //Side3 is equal to base
    printf ("Area Of Triangle_____ %.2f\nPerimeter Of Triangle__ %.2f",at,pt);
}
```

```
printf ("\n\n-----For Rectangle-----\n");
float w,h,ar,pr;
/* where w=Width, h=Height,ar=Area Of Rectangle, pr=Perimeter of Rectangle*/
w=3;
h=6;
ar=h*w;
pr=2*(h+w);
printf ("Area Of Rectangle_____ %.2f\nPerimeter Of Rectangle__ %.2f",ar,pr);
getch ();
return 0;
}
```



## Week 3&4: Basic Decision Constructs and for loop

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which make use of basic decision constructs and for loops.

### Exercise 1:

Write a program that reads 10 positive numbers from the keyboard and determines and displays the sum and average of the numbers.

### Code:

```
/*Program for calculating sum and average (using for loop)*/
#include <stdio.h>
#include <conio.h>
int main ()
{
    int i;
    float n,sum=0,avg; //Avg for average
    printf ("Enter Ten Positive Numbers: ");
    for (i=1;i<=10;i++) //for-loop running 10 times
    {
        scanf ("%f",&n);
        sum=sum+n;
    }
    avg=sum/10;
    printf ("Sum      = %.2f\nAverage = %.2f",sum,avg); //output
    return 0;
}
```

**Exercise 2:**

Print the following series using for loop:

- a. Print numbers from 1 to 100 with increment of 1
- b. Print numbers from 100 to 1 with decrement of 1
- c. Print numbers from 20 to 2 in steps of -2
- d. Print sequence of numbers: 2, 5, 8, 11, 14, 17, 20
- e. Print sequence of numbers: 99, 88, 77, 66, 55, 44, 33, 22, 11, 0

**Code:**

```
#include <stdio.h>
int main ()
{
/*Program for printing numbers from 1 to 100 with increment of 1*/
    int i;
    printf ("Printing numbers from 1 to 100\n");
    for (i=1;i<=100;i++)
    {
        printf ("%d\n",i);
    }

/*Program for printing numbers from 100 to 1 with decrement of 1*/
    printf ("\n\nPrinting numbers from 100 to 1\n");
    for (i=100;i>=1;i--)
    {
        printf ("%d\n",i);
    }

/*Program for printing sequence from 20 to 2 in steps of -2*/
    printf ("\n\nPrinting sequence from 20 to 2 in steps of -2\n");
    for (i=20;i>=2;i-=2)
    {
        printf ("%d\n",i);
    }

/*Program for printing sequence of numbers 2,5,8,11,14,17,20*/
    printf ("\n\nPrinting sequence of numbers\n");
    for (i=2;i<=20;i+=3)
    {
        printf ("%d\n",i);
    }

/*Program for printing sequence of numbers 99,88,77,66,55,44,33,22,11,0*/
    printf ("\n\nPrinting sequence of numbers\n");
    for (i=99;i>=0;i-=11)
    {
```

---

```
        printf ("%d\n",i);  
    }  
  
    return 0;  
}
```

### Exercise 3:

Write a program that reads in five integers and then determines and prints the largest and smallest integers in the group. Use only the techniques you have learnt so far, and make sure that the program only uses three variables. (Hint: use two of the variables to hold the current largest and smallest integers.)

**Code:**

### Exercise 4:

Write a program that reads a number and determines and prints whether it is odd or even. (Hint: use the modulus operator. Any even number is multiple of two, and any multiple of two gives a remainder of zero when divided by two.)

#### Code:

```
/*Program for printing Even or Odd for entered number*/
#include <stdio.h>
int main ()
{
    int n;
    printf ("Please enter any integer number: ");
    scanf ("%d",&n);
    if (n%2==0) //condition for even
        printf ("Entered number is even");
    else // for odd
        printf ("Entered number is odd");
    return 0;
}
```

### Exercise 5:

Write a program which asks the user to enter 10 numbers and prints out the message "even" if the number is even and "divisible by three" if the number is divisible by three.

#### Code:

---

### Exercise 6:

A person invests \$1000.0 in a savings account yielding 5% interest. Assuming that all interest is left on deposit in the account, calculate and print the amount of money in the account at the end of each year for 10 years. Use the following formula for determining these amounts:

$$a = p(1+r)^n$$

where

p is the original investment(i.e. the principal)

r is the annual interest rate

n is the number of years

a is the amount on deposit at the end of the nth year.

### Sample output:

Year	Amount on deposit
1	1050.00
2	1102.50
3	1157.63
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89

Figure 2.1: Sample Output

### Code:

---

## Week 5: while Loops & Sentinel Controlled Repetition

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run more complex programs in C which make use of while loops and sentinel-controlled repetition.

### Exercise 1:

Because of the high price of petrol, you are concerned with the fuel consumption of your car. As a result you have a record of the kilometres driven and litres used for each tank of petrol you purchase. Write a program that will display the kilometres driven, litres used, and consumption (in litres/100km) for each tankful. After processing all the input information, the program should calculate the overall average consumption:

```
Enter the litres used (-1 to end): 57.6
Enter the kilometres driven: 459
The litres/100km for this tank was 12.5
Enter the litres used (-1 to end): 45.3
Enter the kilometres driven: 320
The litres/100km for this tank was 14.2
Enter the litres used (-1 to end): -1
The overall average consumption was: 13.4
```

### Code:

```
#include<stdio.h>

int main()
{
    float lit,km,total;

    printf("Enter the litres used (-1 to end): "); scanf("%f",&lit);
    while(lit>-1)
    {
        printf("Enter the kilometres driven: "); scanf("%f",&km);
```

---

```
    printf("The litres/100km for this tank was %.1f",lit/km*100);
    total += lit/100*km;
    printf("\nEnter the litres used (-1 to end): "); scanf("%f",&lit);
}

printf("\nThe overall average consumption was: %.1f",total);
return 0;
}
```

### Exercise 2:

Write a C program that uses a while loop to calculate and print the sum of the even integers from 2 to 30.

#### Code:

```
#include<stdio.h>

int main()
{
    int no, sum=0;
    for(no=2;no<=30;no++)
        if(no%2==0)
            sum += no;
    printf("\nSum of even numbers from 2 to 30 is %d", sum);
    return 0;
}
```

### Exercise 3:

Write a C program that uses a while loop to calculate and print the product of the odd integers from 3 to 19.

#### Code:





### Exercise 4:

Write a program that reads in a four-digit number, separates the number into its individual digits and prints them separated by three spaces. Thus given the number 4233, the program should print:

4      2      3      3

#### Code:

```
#include<stdio.h>
int main()
{
    int thous,h,hund,ten,ones,n,t;

    printf("Enter the number: ");
    scanf ("%d",&n);
    thous=n/1000;
    h=n%1000;
    hund=h/100;
    t=h%100;
    ten=t/10;
    ones=t%10;
    printf("%d    %d    %d    %d\n", thous, hund, ten, ones);
    return 0;
}
```

### Exercise 5:

Write a C program that will determine if a department store customer has exceeded the credit limit on a charge account. For each customer, the following facts are available:

- a. Account number
- b. Balance at beginning of month
- c. Total of all items charged by the customer this month
- d. Total of all credits applied to the account this month
- e. Allowed credit limit

The program should input each of these facts, calculate the new balance (initial balance + charges - credits) and determine if the new balance exceeds the credit limit. If it does, a suitable message should be displayed.

```
Enter account number (-1 to end): 100
Enter initial balance: 5394.78
Enter total charges: 1000
Enter total credits: 500.00
Enter credit limit: 5500.00
Account: 100
Credit limit: 5500
Balance: 5894.78
```

Credit Limit Exceeded

Enter account number (-1 to end): 200

Enter initial balance: 1000.00

Enter total charges: 123.45

Enter total credits: 321.00

Enter credit limit: 1500.00

Account: 200

Credit limit: 1500

Balance: 802.45

**Code:**

---

## Week 6&7: Functions

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of programs in C which make use of simple functions.

### Exercise 1:

Recall the programme you wrote to implement a simple calculator now rewrite the same programme using functions.

### Code:

```
#include<stdio.h>
#include<conio.h>
int sum(int a,int b)
{
    return a+b;
}
int diff(int a,int b)
{
    return a-b;
}
int div(int a,int b)
{
    return a/b;
}
int prod(int a,int b)
{
    return a*b;
}

int mod(int a,int b)
{
    return a%b;
}
int main()
```

```
{
    int no1,no2,result,i;
    char ch;
    printf("Enter two number : ");scanf("%d%d",&no1,&no2);
    printf("Menu\n");
    printf("Press 1 to add\n");
    printf("Press 2 to Difference\n");
    printf("Press 3 to find product\n");
    printf("Press 4 to divide\n");
    printf("Press 5 to find modulus(first number is dividend and the second number is divisor)\n");
    printf("Enter your choice : ");
    ch = getche();
    switch(ch)
    {
        case '1':
            printf("\nSum of %d and %d is %d",no1,no2,sum(no1,no2));
            break;
        case '2':
            printf("\nDifference of %d and %d is %d",no1,no2,diff(no1,no2));
            break;
        case '3':
            printf("\n %d product %d is %d",no1,no2,prod(no1,no2));

            break;
        case '4':
            if(no2==0)
                printf("\nDivide by zero is not possible!!!");
            else
                printf("\n%d divide by %d is equal to %d",no1,no2,div(no1,no2));
            break;
        case '5':
            printf("\nModulus of %d and %d is %d",no1,no2,mod(no1,no2));
            break;
        default:
            printf("\nWrong option selected!!!");
    }
}

return 0 ;
}
```

## Exercise 2:

Write a program that uses six calls to the function `rand()` to generate six random integer numbers, `num1`, `num2`, `num3`, `num4`, `num5`, and `num6`, and then print them out. `num1` should be in the range 1 to 2 (inclusive), `num2` should be in the range 1 to 100 (inclusive), `num3` should be in the range 0 to 9, `num4` should be in the range 1000 to 1112 (inclusive), `num5` should be in the range -1 to 1, and `num 6` should be in the range -3 to 11.

**Code:**

```

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

int main(void)
{
    int num1,num2,num3,num4,num5,num6;

    printf("\nRandom number in the range 1 to 2 (inclusive)\n\n");
        num1 = rand()%2+1;
    printf("\nnum1 = %d",num1);
    printf("\nRandom number in the range 1 to 100 (inclusive)\n\n");
        num2 = rand()%100+1;
    printf("\nnum2 = %d",num2);

    printf("\nRandom number in the range of 0 to 9\n\n");
        num3 = rand()%10;
    printf("\nnum3 = %d",num3);
    printf("\nRandom number in the range 1000 to 1112 (inclusive)\n\n");
        num4 = rand()%113+1000;
    printf("\nnum4 = %d",num4);
    printf("\nRandom number in the range -1 to 1\n\n");
        num5 = rand()%3-1;
    printf("\nnum5 = %d",num5);
    printf("\nRandom number in the range -3 to 11\n\n");
        num6 = rand()%15-3;
    printf("\nnum6 = %d",num6);
    return 0;
}

```

**Exercise 3:**

A car park charges a \$2.00 minimum fee to park for up to 3 hours, and an additional \$0.50 for each hour or part hour in excess of three hours. The maximum charge for any given 24-hour period is \$10.00. Assume that no car parks for more than 24 hours at a time. Write a C program that will calculate and print the parking charges for each of 3 customers who parked their car in the car park yesterday. The program should accept as input the number of hours that each customer was parked, and output the results in a neat tabular form, along with the total receipts from the three customers:

Car	Hours	Charge
1	1.5	2.00
2	4.0	2.50
3	24.0	10.00
TOTAL	29.5	14.50

Figure 4.1: Sample Output

The program should use the function `calculate_charges` to determine the charge for each customer.

**Code:**

**Exercise 4:**

Implement the following functions. The functions return a real number:

- (a) Function `Celsius` returns the Celsius equivalent of a Fahrenheit temperature (Hint: 0 Celsius is equal to 32 Fahrenheit and 100 Celsius is equal to 212 Fahrenheit).
- (b) Function `Fahrenheit` returns the Fahrenheit equivalent of a Celsius temperature. Use these functions to write a program that prints charts showing the Fahrenheit equivalent of all Celsius temperatures between 0 and 100 degrees, and the Celsius equivalent of all Fahrenheit temperatures between 32 and 212 degrees. Print the output neatly in a table.

**Code:**

**Exercise 5:**

The greatest common divisor of integers  $x$  and  $y$  is the largest integer that divides both  $x$  and  $y$ . Write a recursive function GCD that returns the greatest common divisor of  $x$  and  $y$ . The GCD of  $x$  and  $y$  is defined as follows: If  $y$  is equal to zero, then  $\text{GCD}(x, y)$  is  $x$ ; otherwise  $\text{GCD}(x, y)$  is  $\text{GCD}(y, x \% y)$  where  $\%$  is the remainder operator.

**Code:**



---

## Week 8: Manipulating Arrays

---

### Learning Objectives:

The objective of this exercise is to get you familiar with the collection of data of same type and manipulate it using arrays.

### Exercise 1:

Write a program to declare an integer array of 50 elements.

- A. Write a function `getArray()` to get array input from the user that will be used to initialize the first thirty five (35) elements of the array by getting input from the user. The rest of the 15 entries would be set to zero (0).
- B. Write a function `FindEven()` to find the total numbers of even numbers in the given array.
- C. Write a function `ModifyArray()` to make the each array element to a multiple of four(04).
- D. Write a function named '`FindMin()`' that will find the smallest element in the given array and return the smallest element.

### Code:

```
#include<stdio.h>
#define SIZE 50
void modify(int a[]);
void getarray(int a[],int s);
int FindEven(int a[],int s);
int FindMin(int a[]);
int main()
{
    int array[SIZE]={0},i;
    getarray(array,35);

    modify(array);
    printf("\nTotal number of even value = %d",FindEven(array,SIZE));
    modify(array);
    printf("\nPrinting array after modification:");
```

```
for(i=0 ; i < SIZE ; i++)
    printf("%d",array[i]);
printf("\nMinimum value in the array is %d",FindMin(array));
return 0;
}
void modify(int a[])
{
    int i;
    for(i=0 ; i < SIZE ; i++)
        a[i]*=4;
}
void getarray(int a[],int s)
{
    int i;
    printf("\nEnter %d element : ",s);
    for(i=0;i<s;i++)
    {
        printf("\nEnter element %d : ",i+1); scanf("%d",&a[i]);
    }
}
int FindEven(int a[],int s)
{
    int i,count=0;
    for(i=0;i<s;i++)
        if(a[i]%2==0)
            count++;
    return count;
}
int FindMin(int a[])
{
    int i,min;
    min = a[0];
    for(i=0;i<SIZE;i++)
        if(min>a[i])
            min = a[i];
    return min;
}
```

## Exercise 2:

Develop a program to calculate the basic salary of 5 employees where the hourly rate and the no. of hours worked by each employee are given by the user.

Note: Formula for calculating the basic salary = no. of hours worked \* hourly rate.

### Code:

```
#include<stdio.h>
#include<conio.h>
int main()
```

---

```
{
    float  work = 0.0;
    float  hr = 0.0;
    float  bs[5] = {0.0};
    int i;
    for(i=0;i<5;i++)
    {
        printf("\n enter hours worked and hourly rate for %d employee : ",i+1);
        scanf("%f%f",&work,&hr);
        bs[0] =work+hr;
        printf("\nBasic salary=%f",bs[0]);
    }
    return 0;
}
```

### Exercise 3:

Develop a program to input the marks of 10 (ten) different subjects of a student through the keyboard, find out the aggregate marks and percentage marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.

#### Code:

---

# Week 9: Pointers Notations and Arrays

---

## Learning Objectives:

The objective of this exercise is to get you familiar with pointers, different pointer referencing techniques and array traversing using pointers.

## Exercise 1:

Given the following piece of code

```
int i = 10;
char c= 'A';
double f = 25.5;
int *iptr = &i;
char *cptr = &c;
```

You are expected to display the result of following statements in the given format.

Value of i	Address of i	Value of iptr	Address of iptr	Dereferenced Value of *iptr	Size of iptr	Size of i
Value of c	Address of c	Value of cptr	Address of cptr	Dereferenced Value of *cptr	Size of cptr	Size of c

Note: The format specifier for displaying address of any data item is %x

## Exercise 2:

Implement the swap function to exchange the contents of the two variables using pointer notation for parameter passing.

### Code:

```
#include<stdio.h>

void swap(int *a,int *b)
{
    int hold;
    hold =  *a;
    *a     =  *b;
    *b     =  hold;
return hold;
}
int main()
{
    int x = 10 , y = 20;
    printf("\nBefore swapping");
    printf("\nx=%d,y=%d",x,y);
    swap(&x,&y);
    printf("\nAfter swapping");
    printf("\nx=%d,y=%d",x,y);

return 0;
}
```

## Exercise 3:

Consider the following array of integers:

**Array of integers, named 'arr'**

2	6	-4	8	10	-12	14	16	18	20
num[0]	num[1]	num[2]	num[3]	num[4]	num[5]	num[6]	num[7]	num[8]	num[9]

Implement the following pointer notations to traverse and display the given array 'arr'.

- Printing array using array[i] notation.
- Printing array using ptr[i] notation.
- Printing array using \*(array+i) notation.
- Printing array using \*(ptr+i) notation.
- Printing array using \*ptr notation.

**Code:**

## Week 10: Pointers

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which make use of pointers.

### Exercise 1:

When variables are declared, are they located in memory contiguously? Write a program with the declaration: `char a, b, c, *p, *q, *r;` And print out the locations (addresses) that are assigned to all these variables by your computer. Are the locations in order? If the locations are in order, are they increasing or decreasing? Is the address of each pointer variable divisible by 4? If so, this probably means that each pointer value gets stored in machine word. Double check the results using `sizeof()` for `char` and for `char*`. (Hint: if you want to see addresses printed as decimal numbers rather than hexadecimal, it is usually safe to cast an address as unsigned long and use the `%lu` format).

### Code:

```
#include<stdio.h>
#define SIZE 10
int main()
{
    char a, b, c, *p, *q, *r;
    printf("\nAddress of a = %lu",&a);
    printf("\nAddress of b = %lu",&b);
    printf("\nAddress of c = %lu",&c);
    printf("\nAddress of p = %lu",&p);
    printf("\nAddress of q = %lu",&q);
    printf("\nAddress of r = %lu",&r);
    printf("\nSize of char = %d",sizeof(char));
    printf("\nSize of char* = %d",sizeof(char*));

    return 0;}
```

**Exercise 2:**

Write, compile and execute the program below. Explain why the function swap does not work properly. Using pointers, change the program to make the function swap to work properly.

**Code:**

```
#include <stdio.h>
#include <stdlib.h>
void swap (int a, int b);
int main()
{
    int c = 10, d = 25;
    printf("\nBefore calling the function swap, c=%d and d=%d", c, d);
    swap(c,d);
    printf("\nAfter calling the function swap, c=%d and d=%d\n", c, d); return 0;
}
void swap(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

**Correct Code:**



**Exercise 3:**

Write a program that reads 8 floating point values into an array and then prints out the second, fourth, sixth and eighth members of the array, and the sum of the first, third, fifth and seventh, using pointers to access the members of the array.

**Code:**

---

# Week 11: Strings

---

## Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which make use of strings.

## Exercise 1:

Write a programme which should take an organization name from the user and then print the abbreviation on the screen. E.g. Organization Name is Pakistan Steel and its abbreviation is PS

## Code:

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

int main()
{
    char name[100];
    char abr[50];
    int i=0,j=0;
    printf("Enter organization name : ");
    gets(name);

    abr[j++] = toupper(name[i]);

    for(i=0;name[i]!='\0';i++)
    {
        if(name[i]==' ')
        {
            abr[j] = toupper(name[i+1]);
            j++;
        }
    }
    abr[j] = '\0';
    printf("\n Abbreviation of %s is %s",name,abr);
    return 0; }
```

**Exercise 2:**

Write code for the following functions:

- A. `int strlen(const char *s);` [returns string length]
- B. `char* strcpy(char *s1, const char *s2);` [Copies string s2 into array s1. The value of s1 is returned.]
- C. `char* strcat(char *s1, const char *s2);` [Appends string s2 to array s1. The first character of s2 overwrites the terminating null character of s1. The value of s1 is returned.] Use above function in your program to get and print length of a string, copy str1 to str2 and append str2 at the end of str1. Do not use string.h header file in your programme.

**Code:**

---

## Week 12: Structures

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which make use of structure.

### Exercise 1:

Write a C Program to create the data for some students (roll, name, mark1, mark2, mark3) and then find the total marks for each student and average mark of each student:

#### Code:

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

typedef struct
{
    int rollnum;
    char name[30];
    int marks[3];
}student;

int main()
{
    int i,b,j,t=0,avarage=0;
    student st[4];

    for (i=0;i<4;i++)
    {
        printf("\nStudent roll no: %d",i+1);

        st[i].rollnum=i+1;

        fflush(stdin);

        printf("\nEnter Name:");
        gets(st[i].name);
```

---

```
        for(j=0;j<3;j++)
        {
            printf("\nEnter Marks %d::",j+1);
            scanf("%d",&st[i].marks[j]);
        }
    }
    for (i=0;i<4;i++)
    {
        printf("\nStudent rollno : %d",st[i].rollnum);
        printf("\nName::");
        puts(st[i].name);
        t=0;

        for(j=0;j<3;j++)
        {
            printf("\nMarks : %d",st[i].marks[j]);
            t+=st[i].marks[j];
        }
        avarage = t/3;
        printf("\nTotal : %d",t);
        printf("\nAvarage for student =%d",avarage);
        avarage =0;

    }
    return 0;
}
```

---

## Week 13: Structures

---

### Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which make use of structure.

### Exercise 1:

Write a C Program to create the data for some students (roll, name, mark1, mark2, mark3) and then find the total marks for each student and average mark of each student:

#### Code:

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

typedef struct
{
    int rollnum;
    char name[30];
    int marks[3];
}student;

int main()
{
    int i,b,j,t=0,avarage=0;
    student st[4];

    for (i=0;i<4;i++)
    {
        printf("\nStudent  roll no: %d",i+1);

        st[i].rollnum=i+1;

        fflush(stdin);

        printf("\nEnter Name:");
        gets(st[i].name);
```

---

```
        for(j=0;j<3;j++)
        {
            printf("\nEnter Marks %d::",j+1);
            scanf("%d",&st[i].marks[j]);
        }
    }
    for (i=0;i<4;i++)
    {
        printf("\nStudent rollno : %d",st[i].rollnum);
        printf("\nName::");
        puts(st[i].name);
        t=0;

        for(j=0;j<3;j++)
        {
            printf("\nMarks : %d",st[i].marks[j]);
            t+=st[i].marks[j];
        }
        avarage = t/3;
        printf("\nTotal : %d",t);
        printf("\nAvarage for student =%d",avarage);
        avarage =0;

    }
    return 0;
}
```

# Week 14&15: File Management

---

## Learning Objectives:

The objective of this exercise is to get you to write, compile and run a number of simple programs in C which use file management.

## Exercise 1:

Write a C Program to open a file named "DATA" and write a line of text in it by reading the text from the keyboard.

### Code:

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char ch;

    fp = fopen("data.txt","w");

    if(fp==NULL)
    {
        printf("\nUnable to open file!!!");
        exit(0);
    }
    while((ch =getche())!=13)
    {
        fputc(ch,fp);
    }

    fclose(fp);
    printf("File created successfully!!!");
    return 0;
}
```



---

## Exercise 2:

Write a C Program to read the contents of file 'File1' and paste the contents at the beginning of another file 'File2'.

### Code:

```
#include<stdio.h>
int main()
{
    FILE *source,*target,*temp;
    char file1[20],file2[20];
    char ch;
    printf("\nEnter the source file name to be copied:");
    gets(file1);
    source=fopen(file1,"r");
    if(source==NULL){
        printf("Cannot open %s",file1);
        exit(0);
    }
    printf("\nEnter the destination file name:");
    gets(file2);
    target=fopen(file2,"r");

    if(target==NULL){
        printf("Cannot open %s",file2);
        exit(0);
    }
    temp = fopen("temp.txt","w");
    if(temp==NULL){
        printf("Cannot open %s","temp.txt");
        exit(0);
    }

    while(!feof(source))
    {
        ch=fgetc(source);
        fputc(ch,temp);
    }
    fclose(temp);
    temp = fopen("temp.txt","a");
    if(temp==NULL){
        printf("Cannot open %s","temp.txt");
        exit(0);
    }

    while(!feof(target))
    {
        ch=fgetc(target);
        fputc(ch,temp);
    }
}
```

```
    }
    printf("I am here!!!!");
    fclose(source);
    fclose(target);
    fclose(temp);
    target=fopen(file2,"w");

    if(target==NULL){
        printf("Cannot open %s",file2);
        exit(0);
    }
    temp = fopen("temp.txt","r");
    if(temp==NULL){
        printf("Cannot open %s","temp.txt");
        exit(0);
    }

    while(!feof(temp))
    {
        ch=fgetc(temp);
        fputc(ch,target);

    }
    printf("\nCOMPLETED");
    fclose(temp);
    fclose(target);
    return 0;
}
```

### Exercise 3:

Write a C Program to count the number of characters and spaces in the input supplied from a file.

#### Code: