بِسۡمِ اللّٰہِ الرَّحۡمٰنِ الرَّحِیۡمِ

اللہ کے نام سے شروع جو بڑا مہربان ہمیشہ رحم فرمانے والا ہے۔

Practical Notebook

COMPUTER SCIENCE

10th class.

PUNJAB CURRICULUM AND TEXTBOOK BOARD LAHORE

Authors: **Dr. Muhammad Atif Chattha**
Associate Professor,
Department of Computer Science and IT,
The University of Lahore, Pakistan

Supervision: **Jahanzaib Khan**
Designing: **Aleem Ur Rehman**
Publisher: **Punjab Curriculum and Textbook Board, Lahore**

Director Manuscripts: **Fareeda Sadiq**
Deputy Director (Art & Design): **Ghulam Mohayy- ud-Din**
Layout: **Uzair Ahmad**
Co-oridnator: **Jahanzaib Khan**
Subject Specialist (Computer Science)
Punjab Curriculum and Textbook Board, Lahore

CONTENTS

Printer Date Quantity

CERTIFICATE

Certified that:

Student _____

Class _____ Roll No. _____

School _____

has completed practical work of Physic 9 for

session _____ as per syllabus.

_____ _____

Signature Signature

Science Teacher Head of the Department

Introduction

Computers have become an important part of our daily lives. They can help us to solve several problems ranging from complex mathematical problems and searching on the internet to controlling and operating satellites and rocket launchers. In reality, computers are not very smart on their own. In order to perform all the tasks, they have to be fed a series of instructions by humans which tell them how to behave and perform when faced with a particular type of problem. These series of instructions are known as a *computer program* or *software,* and the process of feeding or storing these instructions in the computer is known as *computer programming*. The person who knows how to write a computer program correctly is known as a *programmer.*

Programming Environment

In order to correctly perform any task, we need to have proper tools. A collection of all the necessary tools for programming makes up a programming environment. It is essential to setup a programming environment before we start writing programs. It works as a basic platform for us to write and execute programs.

Integrated Development Environment (IDE)

A software that provides a programming environment to facilitate programmers in writing and executing computer programs is known as an **Integrated Development Environment (IDE).**

An IDE has a graphical user interface (GUI), meaning that a user can interact with it using windows and buttons to provide input and get output. Some of the many available IDEs for C programming language are: Visual studio, XCode, Code::Blocks, Dev-C++

Compiler

Computers only understand and work in machine language consisting of 0s and 1s. They require the conversion of a program written in a *programming language* to *machine language*, in order to execute it. This is achieved by using a compiler. A **compiler** is a software that is responsible for conversion of a computer program written in some programming language to machine language code.

Programming Basics

Each *programming language* has some primitive building blocks and provides some rules in order to write an accurate program. This set of rules is known as **syntax** of the language. Syntax can be thought of as grammar of a programming language. While programming, if proper syntax or rules of the programming language are not followed, the program does not get compiled. In this case, the compiler generates an error. This kind of errors are called *syntax errors*.

Reserved Words

Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the compiler. These words are known as **reserved words** or **keywords.** If a programmer gives them a definition of his own, it causes a syntax error.

Constants

Constants are the values that cannot be changed by a program e.g. 5, 75.7, 1500 etc. In C language,

primarily we have three types of constants:

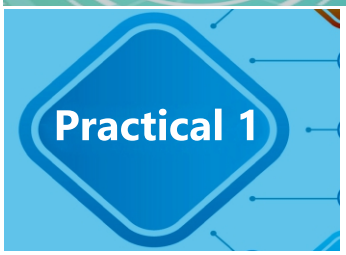
1- **Integer Constants:** These are the numbers without a decimal point e.g. 7, 1256, 30100, 55555, -54, -2349 etc. They can be positive or negative. If the value is not preceded by a sign, it is considered as positive.
2- **Real Constants:** These are the values including a decimal point e.g. 3.14, 15.3333, 75.0, -1575.76, -7941.2345 etc. They can also be positive or negative.
3- **Character Constants:** Any single small case letter, upper case letter, digit, punctuation mark, special symbol enclosed within ' ' is considered a character constant e.g. '5', '7', 'a', 'X', '!', ';' etc.

Variables

A variable is actually a name given to a memory location, as the data is physically stored inside the computer's memory. The value of a variable can be changed in a program. It means that, in a program, if a variable contains value 5, then later we can give it another value that replaces the value 5.

Each variable has a *unique name* called *identifier* and has a *data type*. Data type describes the type of data that can be stored in the variable. C language has different data types such as *int*, *float*, and *char*. The types *int*, *float* and *char* are used to store integer, real and character data respectively. Table shows the matching data types in C language, against different types of data.

Type of Data	Data Type in C Language	Sample Values
Integer	Int	123
Real	Float , Double	23.25
Character	Char	'a'



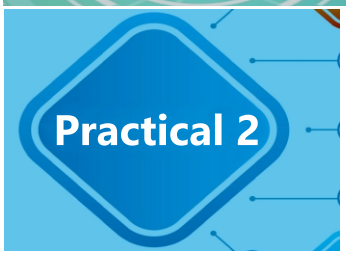
Write a program which uses all types of variables and print their output.

Code:

```
#include<stdio.h>
void main()
{
      int x = 1;
      char c = 'a';
      float y = 2.2;
      double z = 2.32222;
      short a = 8000;
      long b = 983000;
      printf("%d %c %f %lf %hu %ld", x, c, y, z,
      a, b);
}
```

Output:

```
1 a 2.200000 2.32220 8000 983000
```



Write a program to print formatted output using escape sequences.

Code:

```
#include<stdio.h>
void main()
{
      printf("My Name is:
      \tMuhammad\n_\bHassan\a\nI study in class
      10th\t bye\n");
      // single line comment
      /*
      multi
      line
      comment
      */
}
```

Output:

```
My Name is:       Muhammad
Hassan
I study in class 10th       bye
```

Practical 3

Write a program to print the sum and product of two integers and floats.

Code:

```
#include<stdio.h>
void main()
{
      int x, y;
      float a, b;
      printf("Enter two integers: ");
      scanf("%d %d", &x, &y);
      printf("\nEnter two float: ");
      scanf("%f %f", &a, &b);
      printf("Integer:\nSum: %d\nProduct: %d\n",
      x + y, x * y);
      printf("Float:\nSum: %f\nProduct: %f\n", a
      + b, a * b);
}
```

Output:

```
Enter two integers: 2
3

Enter two float: 2.5
3.5
Integer:
Sum:5
Product: 6
Float:
Sum: 6.0000000
Product: 8.750000
```

Practical 4

Write a program to print the difference and division of two floats.

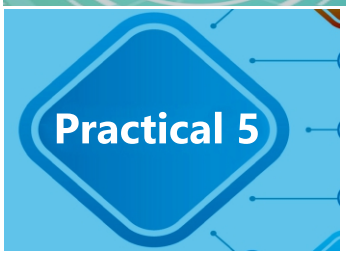
Code:

```
#include<stdio.h>
void main()
{

      float a, b;
      printf("\nEnter two float: ");
      scanf("%f %f", &a, &b);
      printf("Float:\nDifference: %f\nDivision:
      %f\n", a - b, a / b);
}
```

Output:

```
Enter two float: 2.5
2.5
Float:
Difference: 0.000000
Division: 1.000000
```



Write a program to print the use modulus operator on integers.

Code:

```
#include<stdio.h>
void main()
{
      int x = 123;
      printf("%d %d\n", x % 10, x % 100);
}
```

Output:

```
3 23
```

Practical 6

Write a program to print the sum and average of three floats.

Code:

```
#include<stdio.h>
void main()
{
      float x = 1.5, y = 2.5, z = 3.5;
      printf("Sum: %f\nAvg: %f\n", x + y + z, (x
     + y + z) / 3);
}
```

Output:

```
Sum: 7.500000
Avg: 2.500000
```

Practical 7

Write a program to print the cube of a number.

Code:

```
#include<stdio.h>
void main()
{
     float x;
      printf("Enter a number: ");
      scanf("%d", &x);
      printf("\nCube: %d", x * x * x);
}
```

Output:

```
Enter a number: 2
Cube: 8
```

Practical 8

Write a program to print the area of a rectangle.

Code:

```
#include<stdio.h>
void main()
{
      int length, width;
      printf("Enter length and width\n");
      scanf("%d %d", &length, &width);
      printf("Area: %d", length * width);
}
```

Output:

```
Enter length and width
2
4
Area: 8
```

Practical 9

Write a program to print the area of a triangle when three sides are given.

Code:

```
#include<stdio.h>
#include<math.h>
void main()
{
      float a, b, c;
      float s, ans = 0;
      printf("Enter three sides of the
      triangle\n");
      scanf("%f %f %f", &a, &b, &c);
      s = (a + b + c) / 2;
      ans = sqrt(s * (s-a) * (s-b) * (s-c));
      printf("Area of triangle is:%f", ans);
}
```

Output:

```
Enter three sides of the triangle
2
3
4
Area of triangle is: 2.904737
```

Practical 10

Write a program to print the area and circumference of a circle.

Code:

```
#include<stdio.h>
void main()
{
      float radius, circumference, area;
      printf("Enter radius: ");
      scanf("%f", &radius);
      area = 22 * radius * radius / 7;
      circumference= 44 * radius / 7;
      printf("Area: %f\nCircumference: %f\n",
      area, circumference);
}
```

Output:

```
Enter radius: 2.5
Area: 19.642857
Circumference: 15.714286
```

Practical 11

Write a program to print the area and perimeter of square when one side is given

Code:

```
#include<stdio.h>
void main()
{
      int length;
      printf("Enter length\n");
      scanf("%d", &length);
      printf("Area: %d\nPerimeter: %d\n", length
      * length, 4 * length);
}
```

Output:

```
Enter length
3
Area: 9
Perimeter: 12
```

Practical 12

Write a program to print the volume of a cylinder.

Code:

```
#include<stdio.h>
void main()
{
    float radius, height;          of Cylinder
    printf("Enter Radius and Heightof Cylinder\n");
    scanf("%f %f", &radius, &height);
    printf("\nVolume of Cylinder = %f", (22 *
    radius * radius * height / 7));
}
```

Output:

```
Enter Radius and Height of Cylinder
3
4

Volume of Cylinder = 113.142857
```

Practical 13

Write a program to print the surface area and volume of a cube.

Code:

```
#include<stdio.h>
void main()
{
      float side;
      printf("Enter side\n");
      scanf("%f", &side);
      printf("\nSuraface Area of cube =%f\nVolume of
              cube =
      %f\n", side * side, side * side * side);
}
```

Output:

```
Enter side
2

Surface Area of cube    = 4.000000
Volume of cube = 8.000000
```

Practical 14

Write a program to convert temprature from centigrade to fahrenheit.

Code:

```
#include<stdio.h>
void main()
{
      float F;//fahrenheit
      float C;//celcius
      printf("Enter Temperature in Celcius\n");
      scanf("%f", &C);
      F = (C * 9 / 5) + 32;
      printf("Temperature in Fahrenheit = %f", F);
}
```

Output:

```
Enter Temperature in Celcius
36.8
Temperature in Fahrenheit = 98.239998
```

Practical 15

Write a program to convert temperature from Fahrenheit to centigrade.

Code:

```
#include<stdio.h>
void main()
{
      float F;//fahrenheit
      float C;//celcius
      printf("Enter Temperature in  Fahrenheit\n" );
      scanf("%f", &F);
      C = (F - 32) * 5 / 9;
      printf("Temperature in Centigrade = %f", C);
}
```

Output:

```
Enter Temperature in Fahrenheit
98.6
Temperature in Centigrade = 37.000000
```

Practical 16

Write a program to print the distance covered by a car having an average speed.

Code:

```
#include<stdio.h>
void main()
{
      float speed,time,distance;
      printf("Enter Speed and Time in hours\n");
      scanf("%f %f", &speed, &time);
      distance = speed * time;
      printf("Distance covered by car = %f Km
      after %f hours at %f speed", distance,
      time, speed);
}
```

Output:

```
Enter Speed of car and Time in hours
30
3
Distance covered by car = 90.000000 Km after 3.000000 hours at
30.000000 speed
```

Practical 17

Write a program that takes marks in different subjects of a student in class 10. The program should calculate and display the total, obtained marks and percentage.

Code:

```
#include<stdio.h>
void main()
{
      float marks[8],sum=0;
      char name[30], RollNum[6], section;
      int class = 10, i = 0, total;
      printf("Enter Total Marks = ");
      scanf("%d",&total);
      while(i<8)
      {
            printf("Enter marks in subject %d = "
            , i+1);
            scanf("%f",&marks[i]);
            sum+=marks[i]; //sum=sum+marks[i];
            i++;//i =i+1i
      }
      printf("Total marks are %d\nObtained marks
      are = %f\nPercentage is = %f", total, sum,
      100 * sum / total);
}
```

Output:

```
Enter Total Marks = 800
Enter marks in subject 1 = 95
Enter marks in subject 2 = 75
Enter marks in subject 3 = 70
Enter marks in subject 4 = 83
Enter marks in subject 5 = 70
Enter marks in subject 6 = 62
Enter marks in subject 7 = 80
Enter marks in subject 8 = 75
Total marks are = 610.000000
Percentage is = 76.250000
```

Practical 18

Write a program that takes a number as input and displays whether it is even or odd.

Code:

```
#include<stdio.h>
void main()
{
      int n;
      printf("Please Enter a number: ");
      scanf("%d", &n);
      if(n % 2 == 0)
            printf("Even.");
      else
            printf("Odd.");
}
```

Output:

```
Please Enter a number: 3
Odd.
```

Practical 19

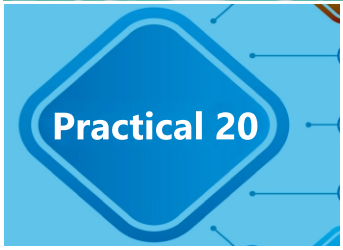
Write a program that takes three numbers as input and displays the largest among them.

Code:

```
#include<stdio.h>
void main()
{
     int x, y, z;
     printf("Enter three numbers\n");
     scanf("%d %d %d", &x, &y, &z);
     if(x>y)
     {
          if(x > z)
               printf("%d is greatest\n", x);
          else
               printf("%d is greatest\n", z);
     }
     else
     {
          if(y > z)
               printf("%d is greatest\n", y);
          else
               printf("%d is greatest\n", z);
     }
}
```

Output:

```
Enter three numbers
2
5
3
5 is greatest
```



Write a program that takes a number as input and prints an error message if the number is not 1 or 2.

Code:

```
#include<stdio.h>
void main()
{
      int x;
      printf("Enter a number\n");
      scanf("%d",&x);
      if(x!=1 && x!=2)
            printf("Error");
}
```

Output:

```
Enter a number
3
Error
```

Practical 21

Write a program that takes three integers as input and displays the largest two.

Code:

```
#include<stdio.h>
void main()
{
      int x, y, z;
      printf("Enter three numbers\n");
      scanf("%d%d%d", &x, &y, &z);
      if(x > y)
      {
            if(z > y)
                  printf("%d and %d are
                  greatest\n", x, z);
            else
                  printf("%d and %d are
                  greatest\n", x, y);
      }
      else if(y > z)
      {
            if(x > z)
                  printf("%d and %d are
                  greatest\n", x, y);
            else
                  printf("%d and %d are
                  greatest\n",z ,y);
      }
      else
      {
            if(x > y)
                  printf("%d and %d are
                  greatest\n",x,z);
            else
                  printf("%d and %d are
                  greatest\n", y, z);
      }
}
```

Output:

```
Enter three numbers
2
7
4
4 and 7 are greatest
```

Practical 22

Write a program that prints the grade of a student.

Code:

```
#include<stdio.h>
void main()
{
      float percentage;
      printf("Enter percentage\n");
      scanf("%f",&percentage);
      if(percentage >= 80)
            printf("A+");
      else if(percentage >= 70)
            printf("A");
      else if(percentage >= 60)
            printf("B");
      else if(percentage >= 50)
            printf("C");
      else if(percentage >= 35)
            printf("D");
      else
            printf("F");
}
```

Output:

```
Enter percentage
67
B
```

Practical 23

Write a program that prints first 10 integers using for loop.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1;
      while(i <= 10)
      {
            printf("%d ", i);
            i++;
      }
}
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

Practical 24

Write a program that prints even numbers from 1 to 10.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1;
      while(i <= 10)
      {
            if(i % 2 == 0)
                  printf("%d ", i);
            i++;
      }
}
```

Output:

```
2 4 6 8 10
```

Practical 25

Write a program that prints odd numbers from 1 to 10.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1;
      while(i <= 10)
      {
            if(i % 2 != 0)
                  printf("%d ", i);
            i++;
      }
}
```

Output:

```
1 3 5 7 9
```

Practical 26

Write a program that takes a number as an input and prints its multiplication table up to 10.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1, num = 0;
      printf("Enter a number\n");
      scanf("%d", &num);
      while(i <= 10)
      {
            printf("%d x %d = %d\n",num,i,num*i);
            i++;
      }
}
```

Output:

```
Enter a number
7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Practical 27

Write a program that takes a number as an input and prints its factorial.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1, num = 0, res = 1;
      printf("Enter a number\n");
      scanf("%d", &num);
      while(i <= num)
      {
            res *= i;
            i++;
      }
      printf("factorial = %d", res);
}
```

Output:

```
Enter a number
5
factorial = 120
```

Practical 28

Write a program that takes two numbers as input and prints the multiplication table of the first number up to the second number.

Code:

```
#include<stdio.h>
void main()
{
      int i=1,num1=0,num2=0,swap=0;
      printf("Enter two  number\n");
      scanf("%d %d",&num1,&num2);
      printf("For table %d is:\n",num1);
      i = 1;
      while(i <= num2)
      {
            printf("%d x %d = %d\n", num1, i,
            num1 * i);
            i++;
      }
     num1++;
}
```

Output:

```
Enter two number
3
5
For table 3 is:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
```

Practical 29

Write a program that prints integers from 10 to 1 (reverse order using loop).

Code:

```
#include<stdio.h>
void main()
{
      int i = 10;
      while(i >= 1)
      {
            printf("%d ", i);
            i--;
      }
}
```

Output:

```
10 9 8 7 6 5 4 3 2 1
```

Practical 30

Write a program that takes input “n from the user and prints the sum of first n integers and average.

Code:

```c
#include<stdio.h>
void main()
{
      int num, sum = 0, i = 1;
      float avg;
      printf("Enter a number\n");
      scanf("%d", &num);
      while(i <= num)
      {
            sum += i;
            i++;
      }
      avg = ((float)sum/(float)num);
      printf("sum = %d \nAverage is = %.2f\n",
      sum, avg);
}
```

Output:

```
Enter a number
4
sum = 10
Average is = 2.50
```

Practical 31

Write a program that takes input “n from the user and prints the first “n” numbers of Fibonacci series.

Code:

```
#include<stdio.h>
void main()
{
      int i1 = 1, i2 = 1, i3 = 010110, i=1, num;
      printf("Enter num\n");
      scanf("%d", &num);
      while(i <= num)
      {
            printf("%d ", i1);
            i3 = i2;
            i2 += i1;
            i1 = i3;
            i++;
      }
}
```

Output:

```
Enter num
7
1 1 2 3 5 8 13
```

Practical 32

Write a program that takes input "n from the user and prints a triangle of * having rows equal to n.

Code:

```c
#include<stdio.h>
void main()
{
	int n, i = 1, j = 1;
	printf("Enter the number of rows\n");
	scanf("%d", &n);
	while(i <= n)
	{
		j = 1;
		while(j <= i)
		{
			printf("*");
			j++;
		}
		printf("\n");
		i++;
	}
}
```

Output:

```
Enter the number of rows
5
*
**
***
****
*****
```

Practical 33

Write a program that creates an array and initializes it with first 10 integers and displays them on screen using loop.

Code:

```
#include<stdio.h>
void main()
{
      int i = 1, a[10];
      while(i <= 10)
      {
            a[i-1]=i;
            i++;
      }
      i = 0;
      while(i < 10)
      {
            printf("%d ", a[i]);
            i++;
      }
}
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

Practical 34

Write a program that takes few numbers from user and prints them in reverse order.

Code:

```
#include<stdio.h>
void main()
{
      int a[5],i=0;
      printf("Enter 5 numbers\n");
      while(i<5)
      {
            scanf("%d",&a[i]);
            i++;
      }
      i=4;
      while(i>=0)
      {
            printf("%d ",a[i]);
            i--;
      }
}
```

Output:

```
Enter 5 numbers
1
2
3
4
5
5 4 3 2 1
```

Practical 35

Write a program that takes few numbers from user and prints the minimum and maximum among them.

Code:

```
#include<stdio.h>
int MAX(int a[],int n)
{
      int mx=a[0],i=0;
      for(i;i<n;i++)
      {
            if(mx<a[i])
                  mx=a[i];
      }
      return mx;
}
int MIN(int a[],int n)
{
      int mn=a[0],i=0;
      for(i;i<n;i++)
      {
            if(mn>a[i])
                  mn=a[i];
      }
      return mn;
}
void main()
{
      int a[5],i=0,max,min;
      printf("Enter 5 numbers\n");
      while(i<5)
      {
            scanf("%d",&a[i]);
            i++;
      }
      max=MAX(a,5);
      min=MIN(a,5);
      printf("Maximum is = %d\nMinimum is = %d
      ",max,min);
}
```

Output:

```
Enter 5 numbers
5
9
2
6
1
Maximum is = 9
Minimum is = 1
```

Practical 36

Write a program that takes 10 numbers as input from user and prints the odd ones.

Code:

```
#include<stdio.h>
void main()
{
      int a[10],i=0;
      while(i<10)
      {
            printf("Enter number %d = ",i+1);
            scanf("%d",&a[i]);
            i++;
      }
      i=0;
      while(i<10)
      {
            if(a[i]%2!=0)
                  printf("%d ",a[i]);
            i++;
      }
}
```

Output:

```
Enter number 1 = 2
Enter number 2 = 3
Enter number 3 = 6
Enter number 4 = 1
Enter number 5 = 78
Enter number 6 = 23
Enter number 7 = 19
Enter number 8 = 23
Enter number 9 = 4
Enter number 10 = 6
3 1 23 19 23
```

Practical 37

Write a program that takes a number as input from user and prints its binary representation.

Code:

```
#include<stdio.h>
void empty(int a[],int n)
{
      int i=0;
      for(i;i<n;i++)
            a[i]=10; //flagged values;
}
void main()
{
      char c;
      int n,a[8],i=0,loop=0;
      printf("Enter a character\n");
      c=getche();
      n=c;
      printf("\n");
      empty(a,8);
      if(c>='0' && n<='9')
      {
            n=n-48;
            while(n>0)
            {
                  a[i]=n%2;
                  n=n/2;
                  i++;
                  loop++;
            }
            i=7;
            while(i>=0)
            {
                  if(a[i]!=10)
                        printf("%d",a[i]);
                  i--;
            }
      }
      else
      {
            while(n>1)
            {
                  a[i]=n%2;
                  n=n/2;
                  i++;
                  loop++;
            }
            a[loop]=1;
            i=7;
            while(i>=0)
            {
                  if(a[i]!=10)
                        printf("%d",a[i]);
                  i--;
            }
      }
}
```

Practical 37

Write a program that takes a number as input from user and prints its binary representation.

Output:

```
Enter a character
9
1001
```

Practical 38

Write a program that takes input from the user. The program passes it to a function which converts it into its ASCII and returns back to the main function. The main function prints its ASCII

Code:

```
#include<stdio.h>
int ascii(char c)
{
      int x=c;
      return x;
}
void main()
{
      char c;
      printf("Enter a char: ");
      c=getche();
      printf("\nASCII is :");
      printf(" %d",ascii(c));
}
```

Output:

```
Enter a char: A
ASCII is: 65
```

Practical 39

Write a program that can take maximum 20 characters from the user. The program saves these characters in an array and prints them on screen as the user is typing.

Code:

```
#include<stdio.h>
void main()
{
      char c[20];
      int i=0;
      printf("Enter characters\n");
      while(i<20)
      {
            c[i]=getche();
            printf("%c",c[i]);
            i++;
      }
}
```

Output:

```
Enter characters
Pprrooggrraammmmiinngg  iiss  ffuunn....
```

Practical 40

Write a program that can take maximum 20 characters from the users as his password. The program then encrypts the password using Caesar cipher and prints it on the screen.

Code:

```
#include<stdio.h>
void main()
{
      char c[20];
      int i=0, size=0;
      printf("Enter your password\n");
      for(i;i<20;i++)
      {
            c[i]=getch();
            if(c[i]=='\r')
                  i=20;
            if(c[i] >= 65 && c[i] <=90 || c[i] >=
            97 && c[i]<=122)
                  c[i]+=1;
      }
      printf("%s",c);
}
```

Output:

```
Enter your password
qsphsbnnjoh jt mpwf
```

Practical 41

Write a program that takes two numbers as input from the user and then passes to a function named sum. The function then returns the sum and the main function prints the output.

Code:

```
#include<stdio.h>
int sum(int x,int y)
{
      return x+y;
}
void main()
{
      int a,b;
      printf("Enter two numbers:\n");
      scanf("%d%d",&a,&b);
      printf("\nSum is = %d\n",sum(a,b));
}
```

Output:

```
Enter two numbers:
3
5

Sum is = 8
```

Practical 42

Write a program that takes two numbers as input from user and passes them to a function to calculate m to the power of n. The function returns the value to main function to print it.

Code:

```
#include<stdio.h>
int power(int x,int y)
{
      int i=1,res=1;
      while(i<=y)
      {
            res*=x;
            i++;
      }
      return res;
}
void main()
{
      int m,n;
      printf("Enter m then n\n");
      scanf("%d%d",&m,&n);
      printf("\nm to the power n is =
      %d\n",power(m,n));
}
```

Output:

```
Enter m then n
4
3

m to the power n is = 64
```

Practical 43

Write a program that takes few numbers as input in an array and then passes them to a function to sort the numbers. The main function prints the sorted array in ascending order.

Code:

```
#include<stdio.h>
void sort(int a[],int n)
{
	int i=0,j=0,swap=0;
	for(i=0; i<n; i++)
	{
		for(j=0; j<n; j++)
		{
			if(a[i]<a[j])
			{
				swap=a[j];
				a[j]=a[i];
				a[i]=swap;
			}
		}
	}
}
void main()
{
	int a[5],i=0;
	printf("Enter 5 numbers: ");
	while(i<5)
	{
		scanf("%d",&a[i]);
		i++;
	}
	sort(a,5);
	i=0;
	while(i<5)
	{
		printf("%d ",a[i]);
		i++;
	}
}
```

Output:

```
Enter 5 numbers: 7
34
87
45
69
7 34 45 69 87
```

Practical 44

Write a program that takes two numbers as input and passes them to a function to calculate GCD. The function returns GCD to main function to print the output.

Code:

```
#include<stdio.h>
int gcd(int x,int y)
{
      int res,rem=1,swap;
      while(rem!=0)
      {
            if(x%y==0)
                  res=y;
            swap=y;
            y=x%y;
            x=swap;
            rem=y;
      }
      return res;
}
void main()
{
      int a,b,swap;
      printf("Enter two numbers: ");
      scanf("%d%d",&a,&b);
      if(a<b)
      {
            swap=a;
            a=b;
            b=swap;
      }
      printf("\nGCD = %d",gcd(a,b));
}
```

Output:

```
Enter two numbers: 45
35

GCD = 5
```

Practical 45

Write a program that takes a five digit integer from the user as input and reverses its presentation. For example, “12345” becomes “54321”.

Code:

```
#include<stdio.h>
void main()
{
      int n;
      printf("Enter a 5 digit num : ");
      scanf("%d",&n);
      while(n!=0)
      {
            printf("%d",n%10);
            n=n/10;
      }
}
```

Output:

```
Enter a 5 digit num : 12345
54321
```

Practical 46

Write a program that takes two numbers as input from the user and performs the basic arithemetic operations on them using modular approach.

Code:

```
#include<stdio.h>
float sum(float x,float y)
{
      return x+y;
}
float sub(float x,float y)
{
      return x-y;
}
float mul(float x,float y)
{
      return x*y;
}
float div(float x,float y)
{
      while(y==0)
      {
            printf("Denominator is zero\nEnter
            denominator again\n");
            scanf("%f",&y);
      }
      return x/y;
}
void main()
{
      int opt;
      float x,y;
      printf("Enter operation you want to
      conduct\n1. SUM\n2. SUB\n3. MUL\n4.DIV\n");
      scanf("%d",&opt);
      printf("\nEnter two numbers : ");
      scanf("%f%f",&x,&y);
      printf("\n");
      if(opt==1)
            printf("Sum = %.2f",sum(x,y));
      else if(opt==2)
            printf("Sub = %.2f",sub(x,y));
      else if(opt == 3)
            printf("Mul = %.2f",mul(x,y));
      else if(opt == 4)
            printf("Div = %.2f",div(x,y));
      else
            printf("Wrong entery");
}
```

Write a program that takes two numbers as input from the user and performs the basic arithemetic operations on them using modular approach.

Output:

```
Enter operation you want to conduct
1. SUM
2. SUB
3. MUL
4.DIV
3

Enter two numbers : 2
5

MUL = 10.00
```

Practical 47

Write a program that takes hourly rate and number of hours of 3 different employees from the user and calculates the salary of each employee.

Code:

```
#include<stdio.h>
void main()
{
      float HourlyRate,HoursWorked[10];
      int i;
      printf("Enter hourly rate : ");
      scanf("%f",&HourlyRate);
      for(i=0; i<3; i++)
      {
            printf("Enter hours worked by
            employee %d = ",i+1);
            scanf("%f",&HoursWorked[i]);
      }
      printf("Printing wage\n");
      for(i=0; i<3; i++)
      {
            printf("Wage of employee %d is
           %.2f\n",i+1,HourlyRate*HoursWorked[i]);
      }
}
```

Output:

```
Enter hourly rate: 200
Enter hours worked by employee 1 = 20
Enter hours worked by employee 2 = 17
Enter hours worked by employee 3 = 23
Printing wage
Wage of employee 1 is 4000.00
Wage of employee 2 is 3400.00
Wage of employee 3 is 4600.00
```

Practical 48

Write a program that can take maximum 20 characters from the user and returns whether the input is palindrome or not.

Code:

```
#include<stdio.h>
#include<string.h>
void main()
{
      char a[20],rev[20];
      int i=0,j=0,flag=1,len;
      gets(a);
      len=strlen(a);
      j=len-1;
      while(i<len)
      {
            rev[i]=a[j];
            i++;
            j--;
      }
      i=0;
      while(i<len)
      {
            if(a[i]!=rev[i])
            flag=0;
            i++;
      }
      if(flag==1)
            printf("Palindrome\n");
      else
            printf("Not a palindrome\n");

}
```

Output:

```
ACDCA
Palindrome
```

Practical 49

Write a program that takes an integer as input from the user and prints whether it is prime or not.

Code:

```
#include<stdio.h>
void main()
{
      int i,n,count=0;
      printf("Enter a number\n");
      scanf("%d",&n);
      if(n==1 || n==0)
      printf("Not a prime\n");
      else
      {
            for(i=1; i<=n; i++)
            {
                  if(n%i==0)
                  count++;
            }
            if(count>2)
                  printf("Not a prime\n");
            else
                  printf("Prime\n");
      }
}
```

Output:

```
Enter a number
7
Prime
```

VIVA QUESTIONS AND ANSWERS

What are numeric Variables?

Numeric variables are those variables that can store numeric values.

What are string variables?

String variables are those variables that can store sequence of characters.

What is constant?

Constant is a quantity whose value cannot be changed; it cannot be changed like a variable. There are numeric constants and string constants.

What is a Numeric Constant?

Numeric constant consists of integers, single-precision or double- precision numbers.

What is a String Constant?

A String constant is a sequence of alphanumeric characters enclosed in double quotation marks. The maximum length of a string constant is 255 characters.

What is a Direct Mode?

When GW-BASIC is loaded, it shows OK message, it is in Direct Mode. In this mode GW-BASIC commands are executed as they are typed. Results are displayed immediately but the commands are lost after execution.

What is an Indirect Mode?

The Indirect mode is used to type the programs. The program loaded in memory is executed by entering RUN command.

What is a Flow Chart?

Flowchart is the pictorial representation of an algorithm. We can present the flow of data in visual form with a Flowchart.

What is an Algorithm?

An Algorithm is a step by step procedure for solving a problem e.g., a recipe for baking a cake is an algorithm?

What are logical errors?

Such errors are called logical errors that are caused in a program due to improper use of symbols and date or wrong use of formula.

What are Syntax Errors?

Syntax errors occur when the program violates one or more grammatical rules of the programing language. These errors are detected at compile time.

What is a Loop?

Loop is a technique to execute a set of statement repeatedly.

What is an array?

Such collection of contiguous memory collections is called array which can store data of same type.

VIVA QUESTIONS AND ANSWERS

What is compiler?

A compiler is a software that is responsible for conversion of a computer program written in some programming language to machine language code.

Define Integrated Development Environment (IDE)?

A software that provides a programming environment which facilitates the programmer in writing and executing computer programs is known as an *Integrated Development Environment (IDE)*.

What is variable initialization?

Assigning value to a variable for the first time is called *variable initialization*. The variable can be initialized at the time of declaration or after declaration.

What are comment statements and why do we add comments?

Comments are the statements that are ignored by the compiler and do not get executed. To include additional information about the program, comments can be used.