VIDUSH SOMANY INSTITUTE OF TECHNOLOGY AND RESEARCH, KADI

KADI SARVA VISHWAVIDYALAYA, GANDHINAGAR

25CE101

FUNDAMENTALS OF PROGRAMMING

LAB MANUAL

SEMESTER – 1

ENROLLMENT NO	
NAME	
BRANCH	

CERTIFICATE

This is to certify that Mr. / Ms. _____

_____ Enrollment No.: _____ has

Satisfactorily completed the course _____ at

Vidush Somany Institute Of Technology & Research,Kadi (Kadi Sarva

Vishwavidyalaya) for _____ Year (B.E.) semester _____

of _____branch in the Academic year _____.

Date of Submission: _____/_____/_____

Faculty Name with Signature **Head of Department**

I N D E X

22	Write a program to check whether entered number is Armstrong or not.	
23	Write a program to check whether entered number is palindrome or not.	
24	Write a program to print factorial of a given number.	
25	Write a program to check whether entered number is prime or not.	
26	Write a program to print Different pattern using For Loop.	
27	Write a program to print 1 to 5 numbers using array.	
28	Write a program to print 1 to 5 reverse numbers using array.	
29	Write a program to find sum and average of five numbers.	
30	Write a program to find maximum and minimum number from given array.	
31	Write a program to find number of positive, negative and zero from given array.	
32	Write a program to find number of odd and even from given array.	
33	Write a program to sort given n number using array.	
34	Write a program to read matrix, display original and transpose of matrix.	
35	Write a program to copy one string to another string.	
36	Write a program to concatenate two strings.	
37	Write a program to find length of given string.	
38	Write a program to find length of given string without using string function.	
39	Write a program to copy one string to another string without using string function.	
40	Write a program to compare two strings.	
41	Write a program to reverse a given string.	
42	Write a program to find given string is palindrome or not.	
43	Write a program to convert a given string into upper case string.	
44	Write a user defined function (UDF) to print whether entered number is odd or even.	
45	Write a program to add first n numbers using user defined function (UDF).	

46	Write a program to find out average of first n numbers using user defined function (UDF).	
47	Write a program to declare structure student having member's grade, name and roll number and access them in various ways.	
48	Write a program using structure to get name, roll number, and marks of a student's of a class and find out who got highest marks. Use concept of structure within structure.	
49	Write a program to create an employee structure having member's name, salary, Get data in employee structure through one function and display data using another function. Use concept of struct and function	
50	Write a program to declare and use pointer variables.	
51	Write a program to swap two values with help of call by value and call by reference.	
52	Write a program to find length of string using pointer and without using string functions.	
53	Write a program to write the characters into file from standard input and then read the characters.	
54	Write a program to write the integers into file from standard input and then read the integers	
55	Write a program that creates the structure of student and Scan the data of n students and store.	
56	Write a program that copies the contents of one file into another.	
57	Write a program that appends the content of file at the end of the other.	

PRACTICAL 1

Write a program to print your address.

Code:

```
#include <stdio.h>
int main() {
   // Printing address directly using printf
   printf("My Address:\n");
   printf("VSITR\n");
   printf("123, MG Road\n");
   printf("Ahmedabad, Gujarat - 380001\n");
   printf("India\n");
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 2

Write a program to perform average of five variables.

Code:

```
#include <stdio.h>
int main() {
   int a, b, c, d, e;
   float avg;

   // Input 5 numbers
   printf("Enter 5 numbers: ");
   scanf("%d %d %d %d %d", &a, &b, &c, &d, &e);

   // Calculate average
   avg = (a + b + c + d + e) / 5.0;

   // Print result
   printf("Average = %.2f\n", avg);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 3

Write a program to print area of circle, rectangle and square.

Code:

```
#include <stdio.h>
#define PI 3.14  // constant value for π

int main() {
   float radius, length, breadth, side;
   float areaCircle, areaRectangle, areaSquare;

   // Circle
   printf("Enter radius of circle: ");
   scanf("%f", &radius);
   areaCircle = PI * radius * radius;

   // Rectangle
   printf("Enter length and breadth of rectangle: ");
   scanf("%f %f", &length, &breadth);
   areaRectangle = length * breadth;

   // Square
   printf("Enter side of square: ");
   scanf("%f", &side);
   areaSquare = side * side;

   // Print results
   printf("Area of Circle = %.2f\n", areaCircle);
   printf("Area of Rectangle = %.2f\n", areaRectangle);
   printf("Area of Square = %.2f\n", areaSquare);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 4

Write a program to convert years into minutes.

Code:

```
#include <stdio.h>

int main() {
   int years;
   long int minutes;

   printf("Enter number of years: ");
   scanf("%d", &years);

   // 1 year = 365 days = 365*24*60 minutes
   minutes = (long int)years * 365 * 24 * 60;

   printf("%d years = %ld minutes\n", years, minutes);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 5

Write a program to perform all the arithmetic operations together in a single program.

Code:

```
#include <stdio.h>

int main() {
   int a, b;
   printf("Enter two numbers: ");
   scanf("%d %d", &a, &b);

   printf("Addition = %d\n", a + b);
   printf("Subtraction = %d\n", a - b);
   printf("Multiplication = %d\n", a * b);
   printf("Division = %.2f\n", (float)a / b);
   printf("Modulus = %d\n", a % b);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 6

Write a program to print a character entered by user.

Code:

```
#include <stdio.h>

int main() {
   char ch;
   printf("Enter a character: ");
   scanf("%c", &ch);

   printf("You entered: %c\n", ch);
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 7

Write a program to convert small letter case to upper letter case.

Code:

```
#include <stdio.h>

int main() {
   char ch;

   printf("Enter a character: ");
   scanf("%c", &ch);

   if (ch >= 'a' && ch <= 'z') {
      ch = ch - 32;   // convert to uppercase
   }

   printf("Uppercase character: %c\n", ch);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 8

Write a program to swap the values of two variables using third variable.

Code:

```
#include <stdio.h>

int main() {
   int a, b, temp;

   printf("Enter two numbers: ");
   scanf("%d %d", &a, &b);

   temp = a;
   a = b;
   b = temp;

   printf("After swapping: a = %d, b = %d\n", a, b);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 9

Write a program to swap the values of two variables without using third variable.

Code:

```
#include <stdio.h>

int main() {
   int a, b;

   printf("Enter two numbers: ");
   scanf("%d %d", &a, &b);

   a = a + b;
   b = a - b;
   a = a - b;

   printf("After swapping: a = %d, b = %d\n", a, b);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 10

Write a program to find maximum and minimum numbers from two numbers by using Conditional operator.

Code:

```
#include <stdio.h>

int main() {
   int a, b, max, min;

   printf("Enter two numbers: ");
   scanf("%d %d", &a, &b);

   max = (a > b) ? a : b;  // conditional operator
   min = (a < b) ? a : b;

   printf("Maximum = %d\n", max);
   printf("Minimum = %d\n", min);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 11

Write a program to demonstrate bitwise operator.

Code:

```
#include <stdio.h>

int main() {
   int a = 5, b = 3;

   printf("a & b = %d\n", a & b);   // AND
   printf("a | b = %d\n", a | b);   // OR
   printf("a ^ b = %d\n", a ^ b);   // XOR
   printf("~a = %d\n", ~a);         // NOT
   printf("a << 1 = %d\n", a << 1); // Left Shift
   printf("a >> 1 = %d\n", a >> 1); // Right Shift

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 12

Write a program to check whether the entered number is odd or even by using if else statement.

Code:

```
#include <stdio.h>

int main() {
   int n;
   printf("Enter a number: ");
   scanf("%d", &n);

   if (n % 2 == 0)
      printf("%d is Even\n", n);
   else
      printf("%d is Odd\n", n);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 13

Write a program to check whether entered character is alphabet, digit or special symbol.

Code:

```
#include <stdio.h>

int main() {
   char ch;
   printf("Enter a character: ");
   scanf("%c", &ch);

   if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z')) {
      printf("'%c' is an Alphabet.\n", ch);
   }
   else if (ch >= '0' && ch <= '9') {
      printf("'%c' is a Digit.\n", ch);
   }
   else {
      printf("'%c' is a Special Symbol.\n", ch);
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 14

Write a program to find whether entered year is leap year or not.

Code:

```
#include <stdio.h>

int main() {
   int year;
   printf("Enter year: ");
   scanf("%d", &year);

   if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))
      printf("%d is a Leap Year\n", year);
   else
      printf("%d is NOT a Leap Year\n", year);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 15

Write a program to check how many days are there in entered month by using switch case.

Code:

```
#include <stdio.h>

int main() {
   int month;
   printf("Enter month number (1-12): ");
   scanf("%d", &month);

   switch (month) {
      case 1: case 3: case 5: case 7: case 8: case 10: case 12:
         printf("31 days\n");
         break;
      case 4: case 6: case 9: case 11:
         printf("30 days\n");
         break;
      case 2:
         printf("28 or 29 days (Leap year check needed)\n");
         break;
      default:
         printf("Invalid month number!\n");
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 16

Write a program to check whether entered character is vowel or consonant by using switch statement.

Code:

```
#include <stdio.h>
int main() {
   char ch;
   printf("Enter a character: ");
   scanf(" %c", &ch);

   switch (ch) {
      case 'a': case 'e': case 'i': case 'o': case 'u':
      case 'A': case 'E': case 'I': case 'O': case 'U':
         printf("%c is a Vowel\n", ch);
         break;
      default:
         printf("%c is a Consonant\n", ch);
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 17

Write a program to get maximum number among three.

Code:

```
#include <stdio.h>

int main() {
   int a, b, c, max;

   printf("Enter three numbers: ");
   scanf("%d %d %d", &a, &b, &c);

   max = a;  // assume a is max
   if (b > max) max = b;
   if (c > max) max = c;

   printf("Maximum = %d\n", max);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 18

Write a program to calculate grade of given marks.

Code:

```
#include <stdio.h>

int main() {
   int marks;
   printf("Enter marks (0-100): ");
   scanf("%d", &marks);

   if (marks >= 90)
      printf("Grade: A\n");
   else if (marks >= 75)
      printf("Grade: B\n");
   else if (marks >= 60)
      printf("Grade: C\n");
   else if (marks >= 40)
      printf("Grade: D\n");
   else
      printf("Grade: F (Fail)\n");

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 19

Write a program to print first 10 integers by using go to statement.

Code:

```
#include <stdio.h>

int main() {
   int i = 1;

start:  // label
   if (i <= 10) {
      printf("%d ", i);
      i++;
      goto start;  // jump back to label
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 20

Write a program to print addition of first n numbers by using go to statement.

Code:

```
#include <stdio.h>

int main() {
   int n, i = 1, sum = 0;

   printf("Enter n: ");
   scanf("%d", &n);

start:
   if (i <= n) {
      sum += i;
      i++;
      goto start;
   }

   printf("Sum of first %d numbers = %d\n", n, sum);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 21

Write a program to find reverse of given numbers. (Example 132-231)

Code:

```
#include <stdio.h>

int main() {
   int n, rev = 0, rem;
   printf("Enter a number: ");
   scanf("%d", &n);

   while (n != 0) {
      rem = n % 10;        // take last digit
      rev = rev * 10 + rem; // build reverse
      n /= 10;             // remove last digit
   }

   printf("Reversed number = %d\n", rev);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 22

Write a program to check whether entered number is Armstrong or not.

Code:

```
#include <stdio.h>

int main() {
   int n, sum = 0, rem, temp;
   printf("Enter a number: ");
   scanf("%d", &n);

   temp = n;
   while (temp != 0) {
      rem = temp % 10;
      sum += rem * rem * rem; // cube of digit
      temp /= 10;
   }

   if (sum == n)
      printf("%d is Armstrong\n", n);
   else
      printf("%d is NOT Armstrong\n", n);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 23

Write a program to check whether entered number is palindrome or not.

Code:

```
#include <stdio.h>

int main() {
   int n, rev = 0, rem, temp;
   printf("Enter a number: ");
   scanf("%d", &n);

   temp = n;
   while (temp != 0) {
      rem = temp % 10;
      rev = rev * 10 + rem;
      temp /= 10;
   }

   if (rev == n)
      printf("%d is Palindrome\n", n);
   else
      printf("%d is NOT Palindrome\n", n);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 24

Write a program to print factorial of a given number.

Code:

```
#include <stdio.h>

int main() {
   int n, i, fact = 1;
   printf("Enter a number: ");
   scanf("%d", &n);

   for (i = 1; i <= n; i++) {
      fact *= i;
   }

   printf("Factorial = %d\n", fact);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 25

Write a program to check whether entered number is prime or not.

Code:

```
#include <stdio.h>

int main() {
   int n, i, isPrime = 1;
   printf("Enter a number: ");
   scanf("%d", &n);

   if (n <= 1)
      isPrime = 0;
   else {
      for (i = 2; i <= n / 2; i++) {
         if (n % i == 0) {
            isPrime = 0;
            break;
         }
      }
   }

   if (isPrime)
      printf("%d is Prime\n", n);
   else
      printf("%d is NOT Prime\n", n);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 26

Write a program to print Different pattern using For Loop.

Code 1:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=i;j++) {
         printf("* ");
      }
      printf("\n");
   }
   return 0;
}
```

Output 1:

Code 2:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=n;i>=1;i--) {
      for(int j=1;j<=i;j++) {
         printf("* ");
      }
      printf("\n");
   }
   return 0;
}
```

Output2:

Code3:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=n-i;j++) printf(" ");
      for(int k=1;k<=i;k++) printf("* ");
      printf("\n");
   }
   return 0;
}
```

Output3:

Code 4:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=n;i>=1;i--) {
      for(int j=1;j<=n-i;j++) printf(" ");
      for(int k=1;k<=i;k++) printf("* ");
      printf("\n");
   }
   return 0;
}
```

Output 4:

Code 5:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=n-i;j++) printf(" ");
      for(int k=1;k<=i;k++) printf("* ");
      printf("\n");
   }
   for(int i=n-1;i>=1;i--) {
      for(int j=1;j<=n-i;j++) printf(" ");
      for(int k=1;k<=i;k++) printf("* ");
      printf("\n");
   }
   return 0;
}
```

Output 5:

Code 6:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=i;j++) {
         printf("%d ", j);
      }
      printf("\n");
   }
   return 0;
}
```

Output 6:

Code 7:

```
#include <stdio.h>
int main() {
   int n=4, num=1;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=i;j++) {
         printf("%d ", num++);
      }
      printf("\n");
   }
   return 0;
}
```

Output 7:

Code 8 :

```
#include <stdio.h>
int main() {
   int n=4;
   for(int i=1;i<=n;i++) {
      for(int j=0;j<i;j++) {
         printf("%c ", 'A'+j);
      }
      printf("\n");
   }
   return 0;
}
```

Output 8 :

Code 9:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=1;i<=n;i++) {
      for(int j=1;j<=n;j++) {
         if(i==1 || i==n || j==1 || j==n)
            printf("* ");
         else
            printf("  ");
      }
      printf("\n");
   }
   return 0;
}
```

Output 9:

Code 10:

```
#include <stdio.h>
int main() {
   int n=5;
   for(int i=0;i<n;i++) {
      int val=1;
      for(int j=0;j<n-i-1;j++) printf(" ");
      for(int k=0;k<=i;k++) {
         printf("%d ", val);
         val = val*(i-k)/(k+1);
      }
      printf("\n");
   }
   return 0;
}
```

Output 10 :

Faculty Signature	
Date and Grade	

PRACTICAL 27

Write a program to print 1 to 5 numbers using array.

Code:

```
#include <stdio.h>

int main() {
   int arr[5] = {1, 2, 3, 4, 5};
   int i;

   for (i = 0; i < 5; i++) {
      printf("%d ", arr[i]);
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 28

Write a program to print 1 to 5 reverse numbers using array.

Code:

```
#include <stdio.h>

int main() {
   int arr[5] = {1, 2, 3, 4, 5};
   int i;

   for (i = 4; i >= 0; i--) {
      printf("%d ", arr[i]);
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 29

Write a program to find sum and average of five numbers.

Code:

```
#include <stdio.h>

int main() {
   int arr[5], i, sum = 0;
   float avg;

   printf("Enter 5 numbers: ");
   for (i = 0; i < 5; i++) {
      scanf("%d", &arr[i]);
      sum += arr[i];
   }

   avg = sum / 5.0;

   printf("Sum = %d\n", sum);
   printf("Average = %.2f\n", avg);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 30

Write a program to find maximum and minimum number from given array.

Code:

```
#include <stdio.h>

int main() {
   int arr[5], i, max, min;

   printf("Enter 5 numbers: ");
   for (i = 0; i < 5; i++) {
      scanf("%d", &arr[i]);
   }

   max = min = arr[0];  // assume first is max & min

   for (i = 1; i < 5; i++) {
      if (arr[i] > max) max = arr[i];
      if (arr[i] < min) min = arr[i];
   }

   printf("Maximum = %d\n", max);
   printf("Minimum = %d\n", min);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 31

Write a program to find number of positive, negative and zero from given array.

Code:

```
#include <stdio.h>

int main() {
   int arr[10], i, pos = 0, neg = 0, zero = 0;

   printf("Enter 10 numbers: ");
   for (i = 0; i < 10; i++) {
      scanf("%d", &arr[i]);

      if (arr[i] > 0) pos++;
      else if (arr[i] < 0) neg++;
      else zero++;
   }

   printf("Positive = %d\n", pos);
   printf("Negative = %d\n", neg);
   printf("Zero = %d\n", zero);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 32

Write a program to find number of odd and even from given array.

Code:

```
#include <stdio.h>

int main() {
   int arr[10], i, odd = 0, even = 0;

   printf("Enter 10 numbers: ");
   for (i = 0; i < 10; i++) {
      scanf("%d", &arr[i]);

      if (arr[i] % 2 == 0)
         even++;
      else
         odd++;
   }

   printf("Even = %d\n", even);
   printf("Odd = %d\n", odd);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 33

Write a program to sort given n number using array.

Code:

```
#include <stdio.h>

int main() {
   int arr[5], i, j, temp;

   printf("Enter 5 numbers: ");
   for (i = 0; i < 5; i++)
      scanf("%d", &arr[i]);

   // Bubble sort
   for (i = 0; i < 5 - 1; i++) {
      for (j = 0; j < 5 - i - 1; j++) {
         if (arr[j] > arr[j + 1]) {
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }

   printf("Sorted numbers: ");
   for (i = 0; i < 5; i++)
      printf("%d ", arr[i]);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 34

Write a program to read matrix, display original and transpose of matrix.

Code:

```
#include <stdio.h>

int main() {
   int a[3][3], t[3][3], i, j;

   printf("Enter 3x3 matrix:\n");
   for (i = 0; i < 3; i++) {
      for (j = 0; j < 3; j++) {
         scanf("%d", &a[i][j]);
         t[j][i] = a[i][j]; // transpose during input
      }
   }

   printf("Original Matrix:\n");
   for (i = 0; i < 3; i++) {
      for (j = 0; j < 3; j++)
         printf("%d ", a[i][j]);
      printf("\n");
   }

   printf("Transpose Matrix:\n");
   for (i = 0; i < 3; i++) {
      for (j = 0; j < 3; j++)
         printf("%d ", t[i][j]);
      printf("\n");
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 35

Write a program to copy one string to another string.

Code:

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

int main() {
   char str1[50], str2[50];

   printf("Enter a string: ");
   gets(str1);

   strcpy(str2, str1); // copy string

   printf("Copied string: %s\n", str2);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 36

Write a program to concatenate two strings.

Code:

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

int main() {
   char str1[50], str2[50];

   printf("Enter first string: ");
   gets(str1);

   printf("Enter second string: ");
   gets(str2);

   strcat(str1, str2); // join strings

   printf("Concatenated string: %s\n", str1);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 37

Write a program to find length of given string.

Code:

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

int main() {
   char str[50];
   printf("Enter a string: ");
   gets(str);

   printf("Length = %d\n", strlen(str));

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 38

Write a program to find length of given string without using string function.

Code:

```
#include <stdio.h>

int main() {
   char str[50];
   int i = 0;

   printf("Enter a string: ");
   gets(str);

   while (str[i] != '\0') {
      i++;
   }

   printf("Length = %d\n", i);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 39

Write a program to copy one string to another string without using string function.

Code:

```
#include <stdio.h>

int main() {
   char str1[50], str2[50];
   int i;

   printf("Enter a string: ");
   gets(str1);

   for (i = 0; str1[i] != '\0'; i++) {
      str2[i] = str1[i];
   }
   str2[i] = '\0'; // null terminate

   printf("Copied string: %s\n", str2);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 40

Write a program to compare two strings.

Code:

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

int main() {
   char str1[50], str2[50];

   printf("Enter first string: ");
   gets(str1);

   printf("Enter second string: ");
   gets(str2);

   if (strcmp(str1, str2) == 0)
      printf("Strings are equal\n");
   else
      printf("Strings are NOT equal\n");

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 41

Write a program to reverse a given string.

Code:

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

int main() {
   char str[50];
   printf("Enter a string: ");
   gets(str);

   strrev(str);  // reverse string (works in Turbo C/Windows compilers)

   printf("Reversed string: %s\n", str);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 42

Write a program to find given string is palindrome or not.

Code:

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

int main() {
   char str[50];
   printf("Enter a string: ");
   gets(str);

   char rev[50];
   strcpy(rev, str);
   strrev(rev);  // reverse

   if (strcmp(str, rev) == 0)
      printf("Palindrome string\n");
   else
      printf("Not Palindrome\n");

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 43

Write a program to convert a given string into upper case string.

Code:

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

int main() {
    char str[50];
    int i;

    printf("Enter a string: ");
    gets(str);

    for (i = 0; str[i] != '\0'; i++) {
        str[i] = toupper(str[i]);  // convert to uppercase
    }

    printf("Uppercase string: %s\n", str);

    return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 44

Write a user defined function (UDF) to print whether entered number is odd or even.

Code:

```
#include <stdio.h>

// function to check odd or even
void checkOddEven(int n) {
   if (n % 2 == 0)
      printf("%d is Even\n", n);
   else
      printf("%d is Odd\n", n);
}

int main() {
   int n;
   printf("Enter a number: ");
   scanf("%d", &n);

   checkOddEven(n);  // function call
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 45

Write a program to add first n numbers using user defined function (UDF).

Code:

```
#include <stdio.h>

// function to return sum of first n numbers
int addN(int n) {
    int sum = 0, i;
    for (i = 1; i <= n; i++)
        sum += i;
    return sum;
}

int main() {
    int n;
    printf("Enter n: ");
    scanf("%d", &n);

    printf("Sum = %d\n", addN(n));
    return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 46

Write a program to find out average of first n numbers using user defined function (UDF).

Code:

```
#include <stdio.h>

// function to calculate average
float avgN(int n) {
   int sum = 0, i;
   for (i = 1; i <= n; i++)
      sum += i;
   return (float)sum / n;
}

int main() {
   int n;
   printf("Enter n: ");
   scanf("%d", &n);

   printf("Average = %.2f\n", avgN(n));
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 47

Write a program to declare structure student having member's grade, name and roll number and access them in various ways.

Code:

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

struct Student {
   char name[50];
   int roll;
   char grade;
};

int main() {
   struct Student s;

   // input data
   printf("Enter name: ");
   gets(s.name);
   printf("Enter roll number: ");
   scanf("%d", &s.roll);
   printf("Enter grade: ");
   scanf(" %c", &s.grade);

   // output data
   printf("\nStudent Info:\n");
   printf("Name: %s\n", s.name);
   printf("Roll: %d\n", s.roll);
   printf("Grade: %c\n", s.grade);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 48

Write a program using structure to get name, roll number, and marks of a student's of a class and find out who got highest marks. Use concept of structure within structure.

Code:

```
#include <stdio.h>

struct Marks {
   int math, sci, eng;
};

struct Student {
   char name[50];
   int roll;
   struct Marks m;
};

int main() {
   struct Student s[3];
   int i, total[3], maxIndex = 0;

   for (i = 0; i < 3; i++) {
      printf("Enter name: ");
      scanf("%s", s[i].name);
      printf("Enter roll: ");
      scanf("%d", &s[i].roll);
      printf("Enter marks (math sci eng): ");
      scanf("%d %d %d", &s[i].m.math, &s[i].m.sci, &s[i].m.eng);

      total[i] = s[i].m.math + s[i].m.sci + s[i].m.eng;
      if (total[i] > total[maxIndex])
         maxIndex = i;
   }

   printf("\nTopper: %s (Roll %d) with total %d\n",
         s[maxIndex].name, s[maxIndex].roll, total[maxIndex]);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 49

Write a program to create an employee structure having member's name, salary, Get data in employee structure through one function and display data using another function. Use concept of struct and function

Code:

```
#include <stdio.h>

struct Employee {
   char name[50];
   float salary;
};

// function to input data
void getData(struct Employee *e) {
   printf("Enter name: ");
   scanf("%s", e->name);
   printf("Enter salary: ");
   scanf("%f", &e->salary);
}

// function to display data
void display(struct Employee e) {
   printf("Employee: %s\n", e.name);
   printf("Salary: %.2f\n", e.salary);
}

int main() {
   struct Employee emp;

   getData(&emp);
   printf("\nEmployee Details:\n");
   display(emp);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 50

Write a program to declare and use pointer variables.

Code:

```
#include <stdio.h>

int main() {
   int a = 10;
   int *p; // pointer

   p = &a; // store address of a

   printf("Value of a = %d\n", a);
   printf("Address of a = %p\n", &a);
   printf("Value of p (address stored) = %p\n", p);
   printf("Value at p = %d\n", *p); // dereference pointer

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 51

Write a program to swap two values with help of call by value and call by reference.

Code:

```
#include <stdio.h>

// Call by Value (won't affect original variables)
void swapByValue(int a, int b) {
   int temp = a;
   a = b;
   b = temp;
   printf("Inside swapByValue: a=%d, b=%d\n", a, b);
}

// Call by Reference (affects original variables)
void swapByRef(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}

int main() {
   int x = 10, y = 20;

   printf("Before Swap: x=%d, y=%d\n", x, y);
   swapByValue(x, y);  // no change in main
   printf("After swapByValue: x=%d, y=%d\n", x, y);

   swapByRef(&x, &y);  // changes in main
   printf("After swapByRef: x=%d, y=%d\n", x, y);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 52

Write a program to find length of string using pointer and without using string functions.

Code:

```
#include <stdio.h>

int main() {
   char str[50];
   char *p;
   int len = 0;

   printf("Enter a string: ");
   gets(str);

   p = str;  // pointer to first char

   while (*p != '\0') { // until null char
      len++;
      p++;
   }

   printf("Length = %d\n", len);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 53

Write a program to write the characters into file from standard input and then read the characters.

Code:

```
#include <stdio.h>

int main() {
   FILE *fp;
   char ch;

   // open file in write mode
   fp = fopen("chars.txt", "w");
   printf("Enter characters (end with #): ");
   while ((ch = getchar()) != '#') {
      fputc(ch, fp);
   }
   fclose(fp);

   // read file
   fp = fopen("chars.txt", "r");
   printf("\nFile contents:\n");
   while ((ch = fgetc(fp)) != EOF) {
      putchar(ch);
   }
   fclose(fp);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 54

Write a program to write the integers into file from standard input and then read the integers.

Code:

```
#include <stdio.h>

int main() {
   FILE *fp;
   int n, i, x;

   fp = fopen("nums.txt", "w");
   printf("How many numbers? ");
   scanf("%d", &n);

   printf("Enter %d numbers: ", n);
   for (i = 0; i < n; i++) {
      scanf("%d", &x);
      fprintf(fp, "%d\n", x); // write integer
   }
   fclose(fp);

   // read back
   fp = fopen("nums.txt", "r");
   printf("\nFile contents:\n");
   while (fscanf(fp, "%d", &x) != EOF) {
      printf("%d ", x);
   }
   fclose(fp);

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 55

Write a program that creates the structure of student and Scan the data of n students and store.

Code:

```
#include <stdio.h>

struct Student {
   char name[50];
   int roll;
   float marks;
};

int main() {
   int n, i;
   struct Student s[50];  // max 50 students

   printf("Enter number of students: ");
   scanf("%d", &n);

   for (i = 0; i < n; i++) {
      printf("\nEnter name: ");
      scanf("%s", s[i].name);
      printf("Enter roll: ");
      scanf("%d", &s[i].roll);
      printf("Enter marks: ");
      scanf("%f", &s[i].marks);
   }

   printf("\n--- Student List ---\n");
   for (i = 0; i < n; i++) {
      printf("Name: %s, Roll: %d, Marks: %.2f\n",
            s[i].name, s[i].roll, s[i].marks);
   }

   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 56

Write a program that copies the contents of one file into another.

Code:

```
#include <stdio.h>

int main() {
   FILE *fp1, *fp2;
   char ch;

   fp1 = fopen("source.txt", "r");
   fp2 = fopen("dest.txt", "w");

   while ((ch = fgetc(fp1)) != EOF) {
      fputc(ch, fp2);
   }

   printf("File copied successfully!\n");

   fclose(fp1);
   fclose(fp2);
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

PRACTICAL 57

Write a program that appends the content of file at the end of the other.

Code:

```
#include <stdio.h>

int main() {
   FILE *fp1, *fp2;
   char ch;

   fp1 = fopen("file1.txt", "r");
   fp2 = fopen("file2.txt", "a");  // append mode

   while ((ch = fgetc(fp1)) != EOF) {
      fputc(ch, fp2);
   }

   printf("File appended successfully!\n");

   fclose(fp1);
   fclose(fp2);
   return 0;
}
```

Output:

Faculty Signature	
Date and Grade	

MCQ QUESTIONS

Q1. In C, which function is used to print your address on the screen?
a) scanf()
b) printf()
c) getch()
d) puts()

Q2. Which header file is required to use printf() function in C?
a) conio.h
b) stdio.h
c) string.h
d) math.h

Q3. If the values of five variables are 10, 20, 30, 40, 50, what will be their average?
a) 25
b) 30
c) 35
d) 40

Q4. Which operator is used for division in C?
a) //
b) /
c) div
d) %

Q5. The formula to calculate area of a circle in C program is:
a) $2 \times \pi \times r$
b) $\pi \times r^2$
c) $\pi \times d^2$
d) $\pi \times r$

Q6. If length = 5 and breadth = 10, what will be the area of rectangle?
a) 50
b) 15
c) 25
d) 100

Q7. 1 year = ? minutes (approx, ignoring leap year).
a) $365 \times 24 \times 60$
b) 365×60
c) $365 \times 12 \times 60$
d) 365×24

Q8. Which operator is used in C to find remainder of division?
a) %
b) /
c) rem
d) div

Q9. To print a single character entered by user, which format specifier is correct in printf()?
a) %s
b) %f
c) %c
d) %d

Q10. Which function is used to take a single character input from user in C?
a) getchar()
b) getch()
c) getche()
d) All of the above

Q11. In ASCII, which value difference exists between lowercase 'a' and uppercase 'A'?
a) 26
b) 32
c) 64
d) 16

Q12. Which function is commonly used in C to convert a lowercase letter to uppercase?
a) lower()
b) convert()
c) toupper()
d) upper()

Q13. Which operator is used to swap two values with the help of a third variable?
a) %
b) =
c) + and –
d) Assignment (=)

Q14. Which technique is used to swap two numbers without using a third variable?
a) Using + and –
b) Using * and /
c) Using bitwise XOR
d) All of the above

Q15. Conditional operator in C is represented as:
a) if-else
b) switch
c) ?:
d) while

Q16. Bitwise AND operator is represented as:
a) &
b) &&
c) |
d) ^

Q17. What will be the result of 5 & 3 (bitwise AND)?
a) 7
b) 5
c) 1
d) 2

Q18. Which keyword is used in C for multiple condition checking with cases?
a) if
b) switch
c) caseif
d) else-if

Q19. If the entered number is 8, which condition is true?
a) 8 % 2 == 1
b) 8 % 2 == 0
c) 8 / 2 == 3
d) 8 % 3 == 2

Q20. Which of the following is NOT a relational operator in C?
a) ==
b) <=
c) =>
d) !=

Q21. A leap year is divisible by:
a) 2
b) 4
c) 100 only
d) 400 only

Q22. Which condition correctly checks for a leap year in C?
a) year % 4 == 0
b) (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)
c) year % 400 == 0 only
d) year % 100 == 0 only

Q23. Which statement is used inside a switch to terminate a particular case?
a) break
b) exit
c) stop
d) continue

Q24. If the entered month is 2, how many days should the program display (ignoring leap year)?
a) 28
b) 29
c) 30
d) 31

Q25. Which characters are considered vowels in C program logic?
a) a, e, i, o, u
b) a, e, i, o, u, A, E, I, O, U
c) only lowercase vowels
d) only uppercase vowels

Q26. What will be the output if three numbers are 10, 25, 15 in a max-of-three program?
a) 10
b) 15
c) 25
d) 50

Q27. If marks = 85, which grade is generally assigned? (Assume A for ≥80, B for 60–79, C for 40–59, F otherwise)
a) A
b) B
c) C
d) F

Q28. Which jump statement can transfer control unconditionally to a labeled statement?
a) continue
b) break
c) goto
d) return

Q29. The goto statement is generally:
a) Recommended for better readability
b) Not recommended in modern programming
c) The only way to make loops
d) Replaces switch-case

Q30. If n=5, the addition of first n numbers using goto will be:
a) 10
b) 15
c) 20
d) 25

Q31. Which loop is most suitable when the number of iterations is known in advance?
a) while
b) do-while
c) for
d) goto

Q32. Reverse of number 132 is:
a) 213
b) 231
c) 321

d) 312

Q33. Armstrong number is defined as:
a) Sum of digits = original number
b) Product of digits = original number
c) Sum of cubes of digits = original number
d) Digits reversed = original number

Q34. Which of the following is an Armstrong number?
a) 123
b) 153
c) 200
d) 321

Q35. A number is called palindrome if:
a) It is divisible by 2
b) It is divisible by 3
c) Its reverse is equal to original number
d) Its square is equal to reverse

Q36. Example of palindrome number is:
a) 121
b) 123
c) 231
d) 456

Q37. Factorial of 5 is:
a) 25
b) 60
c) 120
d) 100

Q38. Prime number is defined as a number:
a) Divisible by 1 only
b) Divisible by 1 and itself only
c) Divisible by 2 always
d) Divisible by 3 always

Q39. Which of the following is NOT a prime number?
a) 2
b) 3
c) 9
d) 7

Q40. In C, nested loops are commonly used for:
a) Simple if condition
b) Pattern printing
c) Function calling
d) Variable declaration

Q41. Which of the following correctly declares an integer array of size 5 in C?
a) int arr(5);
b) int arr[5];
c) array int[5];
d) int[5] arr;

Q42. If an array contains {1,2,3,4,5}, what will be the reverse order?
a) 1 2 3 4 5
b) 5 4 3 2 1
c) 2 3 4 5 1
d) 4 3 2 1 5

Q43. The average of numbers 10, 20, 30, 40, 50 stored in an array is:
a) 20
b) 25
c) 30
d) 40

Q44. Which loop is most commonly used to process all elements of an array?
a) while
b) do-while
c) for
d) switch

Q45. If array = {7, 2, 9}, which is the maximum value?
a) 2
b) 7
c) 9
d) 0

Q46. If array = {7, -3, 0, 4}, how many positive, negative, and zero numbers are there?
a) 2 positive, 1 negative, 1 zero
b) 1 positive, 2 negative, 1 zero
c) 3 positive, 0 negative, 1 zero
d) 2 positive, 2 negative, 0 zero

Q47. If array = {2, 4, 6, 8}, how many odd numbers are there?
a) 0
b) 1
c) 2
d) 4

Q48. Bubble sort is an algorithm used for:
a) Searching
b) Sorting
c) Multiplying
d) Reversing

Q49. After sorting {5, 1, 4, 2} in ascending order, the array becomes:
a) 5 4 2 1
b) 1 2 4 5
c) 4 2 1 5
d) 2 1 5 4

Q50. Transpose of a matrix is obtained by:
a) Reversing all rows
b) Swapping rows with columns
c) Reversing all columns
d) Adding diagonal elements

Q51. Which library in C provides string handling functions like strcpy(), strlen(), strcmp()?
a) stdio.h
b) conio.h
c) string.h
d) math.h

Q52. The function used to copy one string into another is:
a) strcat()
b) strcpy()
c) strcmp()
d) strlen()

Q53. Which function is used to join two strings in C?
a) strcat()
b) strcpy()
c) strcmp()
d) strrev()

Q54. If str = "hello", what is the length of the string (excluding '\0')?
a) 6
b) 5
c) 4
d) 0

Q55. Which function is used to calculate string length in C?
a) strlen()
b) strlength()
c) len()
d) size()

Q56. Without using string functions, how is the length of a string usually calculated?
a) By counting characters until '\0' is found
b) By checking array size
c) By using a for loop with fixed length
d) By calling printf()

Q57. Which function compares two strings in C?
a) strcmp()
b) strcpy()
c) strcat()
d) strcomp()

Q58. Which function is used to reverse a string?
a) strrev()
b) reverse()
c) strback()
d) revstr()

Q59. A string is palindrome if:
a) Its characters are sorted
b) It reads same forward and backward
c) It contains vowels only
d) It contains spaces

Q60. Which function is used to convert lowercase string to uppercase?
a) strupr()
b) strtoupper()
c) upper()
d) convertupper()

Q61. In C, a user-defined function (UDF) is created using which keyword?
a) func
b) function
c) void / return type
d) define

Q62. Which of the following is a correct function prototype in C?
a) int add(int, int);
b) add(int, int) int;
c) function add(int, int);
d) int add;

Q63. A UDF to check odd/even number will generally return:
a) The number itself
b) A string value
c) 1 or 0 (true/false)
d) The square of number

Q64. Which method is used to pass arguments from calling function to called function?
a) Call by value
b) Call by reference
c) Both a and b
d) None

Q65. The keyword struct in C is used to:
a) Declare arrays
b) Declare a new data type that groups variables
c) Declare loops
d) Define constants

Q66. If a structure student has members name, rollno, and grade, how are they accessed (if variable is s1)?
a) s1.name, s1.rollno, s1.grade
b) s1->name, s1->rollno, s1->grade
c) struct.s1.name
d) s1:name

Q67. Which operator is used with pointers to access structure members?
a) . (dot)
b) -> (arrow)
c) * (asterisk)
d) & (ampersand)

Q68. A structure within a structure is called:
a) Nested structure
b) Array of structure
c) Pointer structure
d) Complex structure

Q69. If an employee structure has members (name, salary), which function would be used to input data?
a) scanf()
b) printf()
c) strcpy()
d) strlen()

Q70. Which of the following is NOT true about structures in C?
a) They group different data types together
b) They can be passed to functions
c) They cannot contain arrays
d) They can contain another structure

Q71. A pointer in C stores:
a) Actual value of a variable
b) Address of a variable
c) Both value and address
d) Only integers

Q72. Which operator is used to get the address of a variable?
a) *
b) ->
c) &
d) %

Q73. Which operator is used to access the value stored at an address?
a) *
b) &
c) ->
d) %

Q74. Call by value means:
a) Copy of variable is passed to function
b) Address of variable is passed
c) No value is passed
d) Function directly changes original value

Q75. Call by reference means:
a) Copy of variable is passed
b) Address of variable is passed
c) Function cannot modify the variable
d) Value cannot be used in function

Q76. Which header file is required for file operations in C?
a) stdlib.h
b) stdio.h
c) file.h
d) conio.h

Q77. Which function is used to open a file in C?
a) fopen()
b) open()
c) fileopen()
d) fcreate()

Q78. Which mode is used in fopen() to append data to a file?
a) "w"
b) "a"
c) "r"
d) "wa"

Q79. Which function is used to read a character from a file?
a) putc()
b) getc()
c) read()
d) freadc()

Q80. Which function is used to close a file in C?
a) fclose()
b) close()
c) fileclose()
d) endfile()

Q81. Which mode in fopen() is used to create a new file for writing?
a) "r"
b) "w"
c) "a"
d) "rw"

Q82. If a file opened in "w" mode already exists, what happens?
a) File remains unchanged
b) File is deleted
c) File contents are erased and replaced
d) New file is created with different name

Q83. To read integers from a file, which function is commonly used?
a) fscanf()
b) printf()
c) freadint()
d) getc()

Q84. Which function is used to write a string into a file?
a) puts()
b) fprintf()
c) fwrite()
d) both b and c

Q85. In file handling, EOF stands for:
a) End Of File
b) Error On File
c) Extra Output Function
d) Empty Open File

Q86. Which function is used to detect end of file in C?
a) eof()
b) feof()
c) endfile()
d) fclose()

Q87. Which function is used to copy content of one file into another character by character?
a) putc() with getc()
b) fwrite() with fread()
c) fcopy()
d) strcat()

Q88. Which file mode is used to open a file for both reading and writing?
a) "rw"
b) "r+"
c) "wr"
d) "a+"

Q89. Which function is used to append data from one file into another?
a) fappend()
b) fprintf() with "a" mode
c) fwrite() with "a+" mode
d) both b and c

Q90. What is the default location of file creation in C if no path is specified?
a) C:\Windows\System32
b) Same directory where program runs
c) C:\
d) Desktop

Q91. In C, the size of a character variable is usually:
a) 2 bytes
b) 4 bytes
c) 1 byte
d) Depends on compiler

Q92. Which header file is required for toupper() and tolower() functions?
a) ctype.h
b) string.h
c) stdio.h
d) conio.h

Q93. Which function is used to dynamically allocate memory in C?
a) malloc()
b) calloc()
c) realloc()
d) All of the above

Q94. Which of the following is the correct syntax to declare a pointer to integer?
a) int p;
b) int *p;
c) pointer p;
d) int &p;

Q95. Which of the following is not a loop in C?
a) for
b) while
c) repeat-until
d) do-while

Q96. Which operator is used to access array elements?
a) . (dot)
b) -> (arrow)
c) [] (subscript)
d) *

Q97. Which of the following keywords is used to define a constant in C?
a) const
b) #define
c) final
d) both a and b

Q98. In C, which escape sequence is used for a new line?
a) /n
b) \n
c) \n
d) newline

Q99. Which of the following is not a valid C identifier?
a) _value
b) 1value
c) value1
d) value_1

Q100. Which concept allows breaking a program into smaller reusable parts?
a) Arrays
b) Loops
c) Functions
d) Pointers

PRACTICE PRACTICAL (SELF STUDY)

1. Write a program to print your college name and city.
2. Write a program to print your date of birth in DD/MM/YYYY format.
3. Write a program to calculate the sum of three subject marks.
4. Write a program to calculate the average of 3 numbers.
5. Write a program to calculate the perimeter of circle, rectangle, and square.
6. Write a program to calculate the volume of a cube and cuboid.
7. Write a program to convert days into years and weeks.
8. Write a program to convert hours into seconds.
9. Write a program to perform modulus, increment, and decrement operations together.
10. Write a program to calculate simple interest.
11. Write a program to calculate compound interest.
12. Write a program to read a digit and print its ASCII value.
13. Write a program to print the ASCII value of all alphabets.
14. Write a program to convert uppercase character to lowercase.
15. Write a program to convert lowercase character to uppercase.
16. Write a program to swap three variables using a temporary variable.
17. Write a program to swap two variables using arithmetic operators.
18. Write a program to swap two variables using bitwise XOR.
19. Write a program to find the maximum among two numbers using if-else.
20. Write a program to find the minimum among two numbers using if-else.
21. Write a program to demonstrate logical operators.
22. Write a program to demonstrate relational operators.
23. Write a program to check whether a number is divisible by 5.
24. Write a program to check whether a number is divisible by 7 and 11.
25. Write a program to check whether a character is vowel, consonant, or digit.
26. Write a program to check whether a number is positive, negative, or zero.
27. Write a program to check whether a year is a century year or not.
28. Write a program to check whether a year is divisible by 400 or not.
29. Write a program to check whether a month number has 28, 30, or 31 days using switch.
30. Write a program to check whether an alphabet is uppercase or lowercase using switch.
31. Write a program to print the maximum among three numbers.
32. Write a program to print the minimum among three numbers.
33. Write a program to calculate percentage of marks.
34. Write a program to calculate grade of student using nested if.
35. Write a program to calculate grade of student using switch case.
36. Write a program to print first 15 natural numbers using goto.
37. Write a program to print first 20 odd numbers using goto.
38. Write a program to print sum of first 10 numbers using goto.
39. Write a program to print product of first 5 numbers using goto.
40. Write a program to reverse the digits of a number using while loop.
41. Write a program to find the sum of digits of a number.
42. Write a program to check whether a number is a Perfect Number.
43. Write a program to check whether a number is a Neon Number.
44. Write a program to check whether a number is a Strong Number.
45. Write a program to check whether a number is an Automorphic Number.
46. Write a program to print the factors of a number.
47. Write a program to print all prime numbers between 1 to 100.

48. Write a program to print all Armstrong numbers between 1 to 1000.
49. Write a program to print the first 10 Fibonacci numbers.
50. Write a program to print the factorial of numbers from 1 to 10.
51. Write a program to print star pattern in triangle form.
52. Write a program to print star pattern in reverse triangle form.
53. Write a program to print star pattern in pyramid form.
54. Write a program to print number pyramid pattern.
55. Write a program to print 1 to 10 numbers using array.
56. Write a program to print 10 to 1 numbers using array.
57. Write a program to print squares of first 10 numbers using array.
58. Write a program to calculate sum of 10 numbers using array.
59. Write a program to calculate average of n numbers using array.
60. Write a program to find the second largest number from an array.
61. Write a program to find the second smallest number from an array.
62. Write a program to count even and odd numbers from an array.
63. Write a program to count positive and negative numbers from an array.
64. Write a program to count zeros in an array.
65. Write a program to sort an array in ascending order.
66. Write a program to sort an array in descending order.
67. Write a program to merge two arrays.
68. Write a program to reverse an array.
69. Write a program to add two matrices.
70. Write a program to subtract two matrices.
71. Write a program to multiply two matrices.
72. Write a program to print transpose of a matrix.
73. Write a program to check whether a matrix is symmetric or not.
74. Write a program to copy one string into another using strcpy().
75. Write a program to concatenate two strings using strcat().
76. Write a program to find length of string using strlen().
77. Write a program to find length of string without using strlen().
78. Write a program to compare two strings using strcmp().
79. Write a program to reverse a string using strrev().
80. Write a program to reverse a string without using strrev().
81. Write a program to check whether two strings are equal or not.
82. Write a program to check whether two strings are anagrams.
83. Write a program to count vowels and consonants in a string.
84. Write a program to count spaces in a string.
85. Write a program to remove spaces from a string.
86. Write a program to convert a string into uppercase.
87. Write a program to convert a string into lowercase.
88. Write a program to check whether a string is palindrome.
89. Write a program to check whether a string is pangram.
90. Write a program to create a UDF to calculate factorial.
91. Write a program to create a UDF to check prime number.
92. Write a program to create a UDF to check Armstrong number.
93. Write a program to create a UDF to find sum of digits.
94. Write a program to declare a structure Book with title, author, and price.
95. Write a program to declare a structure Car with name, model, and price.
96. Write a program to declare a structure Teacher with name, department, and salary.
97. Write a program to use pointer to print elements of an array.

98. Write a program to demonstrate pointer to pointer.
99. Write a program to swap two variables using call by reference only.
100. Write a program to find length of string using recursion and pointers.

PROJECT

Project Title: Student Result Management System

Objective

The purpose of this project is to build a small C program that can store and manage student results. Students will practice **arrays, structures, functions, and conditional statements**.

Requirements

1. **Input Section**
   - Ask the user how many students' data they want to enter.
   - For each student, input:
     - Roll number
     - Name
     - Marks in 3 subjects
2. **Processing Section**
   - Calculate:
     - Total Marks = sum of 3 subjects
     - Percentage = (Total Marks ÷ 3)
     - Grade = based on percentage (use conditions):
       - ≥ 80 → A
       - 60–79 → B
       - 40–59 → C
       - < 40 → Fail
3. **Output Section**
   - Display all student records in **tabular form** with:
     - Roll No | Name | Marks | Total | Percentage | Grade
4. **Search Feature**
   - Allow the user to enter a roll number.
   - Display only that student's record if found.
   - Otherwise, print "Record not found."

Expected Sample output:

```
Enter number of students: 2
Enter details of Student 1:
Roll No: 101
Name: Ravi
Marks (3 subjects): 75 80 65
Enter details of Student 2:
Roll No: 102
Name: Meena
Marks (3 subjects): 50 60 55
---------------------------------------------
Roll | Name   | Total | Percentage | Grade
---------------------------------------------
101  | Ravi   | 220   | 73.33%     | B
102  | Meena  | 165   | 55.00%     | C
---------------------------------------------
Enter roll number to search: 101
Result of Roll No 101 → Name: Ravi | Total: 220 | Percentage: 73.33 | Grade: B
```

Solution:

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

// structure for student
struct Student {
    int roll;
    char name[50];
    int marks[3];
    int total;
    float percentage;
    char grade;
};

// function to calculate total, percentage and grade
void calculateResult(struct Student *s) {
    s->total = 0;
    for (int i = 0; i < 3; i++) {
        s->total += s->marks[i];
    }
    s->percentage = (float)s->total / 3.0;

    if (s->percentage >= 80)
        s->grade = 'A';
    else if (s->percentage >= 60)
        s->grade = 'B';
    else if (s->percentage >= 40)
        s->grade = 'C';
    else
        s->grade = 'F'; // F for Fail
}

// function to display all students
void displayAll(struct Student s[], int n) {
    printf("\n---------------------------------------------\n");
    printf("Roll | Name       | Total | Percentage | Grade\n");
    printf("---------------------------------------------\n");
    for (int i = 0; i < n; i++) {
        printf("%4d | %-10s | %5d | %9.2f | %c\n",
               s[i].roll, s[i].name, s[i].total, s[i].percentage, s[i].grade);
    }
    printf("---------------------------------------------\n");
}

// function to search by roll number
void searchStudent(struct Student s[], int n, int roll) {
    int found = 0;
    for (int i = 0; i < n; i++) {
        if (s[i].roll == roll) {
```

```
            printf("\nResult of Roll No %d:\n", roll);
            printf("Name: %s\n", s[i].name);
            printf("Total: %d\n", s[i].total);
            printf("Percentage: %.2f\n", s[i].percentage);
            printf("Grade: %c\n", s[i].grade);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("\nRecord not found for Roll No %d\n", roll);
    }
}

int main() {
    int n;
    printf("Enter number of students: ");
    scanf("%d", &n);

    struct Student s[n];

    // input details
    for (int i = 0; i < n; i++) {
        printf("\nEnter details of Student %d:\n", i + 1);
        printf("Roll No: ");
        scanf("%d", &s[i].roll);
        printf("Name: ");
        scanf("%s", s[i].name);
        printf("Enter 3 subject marks: ");
        for (int j = 0; j < 3; j++) {
            scanf("%d", &s[i].marks[j]);
        }
        calculateResult(&s[i]); // calculate result for this student
    }

    // display all records
    displayAll(s, n);

    // search functionality
    int roll;
    printf("\nEnter roll number to search: ");
    scanf("%d", &roll);
    searchStudent(s, n, roll);

    return 0;
}
```

NOTES