

**COMPUTER PROGRAMMING AND**  
**NUMERICAL METHODS**

**JADAVPUR UNIVERSITY**



**NAME – ARKO DASGUPTA**

**ROLL – 002310501085**

**DEPARTMENT**

**COMPUTER SCIENCE AND ENGINEERING**

**(BCSE UG-1 2023-24)**

**(GROUP – A3)**

## **INDEX**

<b>SL. NO</b>	<b>ASSIGNMENT</b>	<b>PAGE</b>	<b>SIGNATURE</b>
1	Assignment 1 (Console I/O and Conditional Statements – A)	3	
2	Assignment 2 (Console I/O and Conditional Statements – B)	16	
3	Assignment 3 (Loops)	33	
4	Assignment 4 (Arrays)	79	
5	Assignment 5 (Functions and Pointers) (Till Q10)	105	

# Assignment 1

1. Write a C program that reads two values from the keyboard, swaps their values, and prints out the result.

Ans: CODE—

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

```
int main()
```

```
{
```

```
    float a, b;
```

```
    printf("Enter two numbers: ");
```

```
    scanf("%f %f", &a, &b);
```

```
    printf("Numbers before swapping: a = %.2f, b = %.2f\n", a, b);
```

```
    float tmp = a;
```

```
    a = b, b = tmp;
```

```
    printf("Numbers after swapping: a = %.2f, b = %.2f\n", a, b);
```

```
    return 0;
```

```
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q1.exe
Enter two numbers: 5 8
Numbers before swapping: a = 5.00, b = 8.00
Numbers after swapping: a = 8.00, b = 5.00
```

**2. The length and breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area and perimeter of the rectangle, and the area and circumference of the circle.**

**Ans: CODE—**

```
#include <stdio.h>

#define PI 3.14

int main()
{
    float r, l, b;

    printf("Enter radius of circle, length and breadth of rectangle: ");
    scanf("%f %f %f", &r, &l, &b);

    // rectangle
    printf("Perimeter of rectangle: %.2f\n", 2 * (l + b));
    printf("Area of rectangle: %.2f\n", l * b);

    // area
    printf("Circumference of circle: %.2f\n", 2 * PI * r);
    printf("Area of circle: %.2f\n", PI * r * r);

    return 0;
}
```

**OUTPUT—**

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q2.exe
Enter radius of circle, length and breadth of rectangle: 3 4 5
Perimeter of rectangle: 18.00
Area of rectangle: 20.00
Circumference of circle: 18.84
Area of circle: 28.26
```

**3. If a three-digit integer is input through the keyboard, write a program to calculate the sum of its digits. (Hint: Use the modulo operator ‘%’)**

**Ans:** CODE—

```
#include <stdio.h>

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

    int n1 = n;

    if (n < 0)
        n = -n;

    while (n)
    {
        s += (n % 10);
        n /= 10;
    }

    printf("The sum of digits of %d is %d\n", n1, s);

    return 0;
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q3.exe
Enter number: 12345
The sum of digits of 12345 is 15
```

**4. An integer is entered as an input through the keyboard. Write a program to find out whether it is an odd number or an even number.**

**Ans:** CODE—

```
#include <stdio.h>

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

    if (n % 2 == 0)
    {
        printf("%d is even", n);
    }
    else
    {
        printf("%d is odd", n);
    }

    return 0;
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q4.exe
Enter number: 67
67 is odd
```

**5. Input two integer numbers and divide the larger number by the smaller one. Then display the result using printf() function as a fractional number first and then as a real valued number.**

**(Example: 9 divided by 5 shall yield “9/5” and “1.8” respectively.)**

**Ans:** CODE—

```
#include <stdio.h>

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

    if (a > b)
        l = a, s = b;

    else
        l = b, s = a;

    printf("Fractional form: %d / %d\n", l, s);
    printf("Decimal form: %.1f\n", (float)l / s);

    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q5.exe
Enter two numbers: 9 5
Fractional form: 9/5
Decimal form: 1.8
```

**6. Write a C program to find the maximum and minimum of three numbers.**

**Ans:** CODE –

```
#include <stdio.h>

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

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

    float arr[3];
    arr[0] = a, arr[1] = b, arr[2] = c;
    float max = arr[0], min = arr[0];
    for (int i = 1; i <= 2; ++i)
    {
        if (arr[i] > max)
            max = arr[i];
        if (arr[i] < min)
        {
            min = arr[i];
        }
    }

    printf("Maximum number: %.2f\n", max);
    printf("Minimum number: %.2f\n", min);
    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q6.exe
Enter three numbers: 9 14 5
Maximum number: 14.00
Minimum number: 5.00
```



**7. Accept three integer numbers and find their average. Next display which numbers are below and above the average value.**

**Ans:**    CODE—

```
#include <stdio.h>

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

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

    float avg = (float) (a + b + c) / 3;
    printf("Average value: %.2f\n", avg);

    // a
    {
        if (a > avg)
            printf("%d is above average\n", a);
        else if (a == avg)
            printf("%d is same as average\n", a);
        else
            printf("%d is below average\n", a);
    }

    {
        if (b > avg)
            printf("%d is above average\n", b);
        else if (b == avg)
            printf("%d is same as average\n", b);
        else
            printf("%d is below average\n", b);
    }
}
```

```

    }
    {
        if (c > avg)
            printf("%d is above average\n", c);
        else if (c == avg)
            printf("%d is same as average\n", c);
        else
            printf("%d is below average\n", c);
    }
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q7.exe
Enter three numbers: 12 6 19
Average value: 12.33
12 is below average
6 is below average
19 is above average

```

**8. Temperature of a city in Fahrenheit degrees is input through the keyboard. Write a program to convert this temperature into Centigrade degrees.**

**Ans:** CODE—

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

```
int main()
```

```
{
```

```
    float fahrenheit;
```

```
    printf("Enter temperature in Fahrenheit: ");
```

```
    scanf("%f", &fahrenheit);
```

```
float centigrade = (5.0 / 9.0) * (fahrenheit - 32.0);

printf("Temperature in centigrade: %.2f\n", centigrade);

return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q8.exe
Enter temperature in Fahrenheit: 212
Temperature in centigrade: 100.00
```

**9. Write a C program which accepts basic salary as input and prints the gross salary, which is sum of the basic, dearness allowance (60% of basic salary), and house rent allowance (15% of basic salary).**

**Ans:** CODE –

```
#include <stdio.h>

int main()
{
    float salary;

    printf("Enter salary: ");

    scanf("%f", &salary);

    float gross = salary + (0.15 * salary) + (0.6 * salary);

    printf("Gross salary: %.2f\n", gross);

    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q9.exe
Enter salary: 50000
Gross salary: 87500.00
```

**10. A cashier has currency notes of denominations 1, 5, 10, 50 and 100. Write a C program which accepts an amount to be withdrawn, and prints the total number of currency notes of each denomination the cashier will have to give to the withdrawer.**

**Ans:** CODE –

```
#include <stdio.h>

int main()
{
    int amt;

    printf("Enter amount to be withdrawn: Rs ");
    scanf("%d", &amt);

    printf("Denominations:\n");

    printf("100s: %d\n", amt / 100);
    printf("50s %d\n", (amt % 100) / 50);
    printf("10s: %d\n", ((amt % 100) % 50) / 10);
    printf("5s: %d\n", (((amt % 100) % 50) % 10) / 5);
    printf("1s: %d\n", (((amt % 100) % 50) % 10) % 5));

    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q10.exe
Enter amount to be withdrawn: Rs 578
Denominations:
100s: 5
50s 1
10s: 2
5s: 1
1s: 3
```

**11. If the marks obtained by a student in five different subjects are input 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. Input error should be checked.**

**Ans:** CODE –

```
#include <stdio.h>

#define TOTAL 500

int main()
{
    float marks, sum = 0.0;
    int i = 0;
    printf("Enter marks of student in 5 subjects:\n");
    while (i < 5)
    {
        printf("subject %d: ", i + 1);
        scanf("%f", &marks);

        if (marks > 100)
        {
            printf("Marks cannot be greater than total marks 100\nEnter again\n");
        }
        else
        {
            sum += marks;
            i += 1;
        }
    }

    printf("Total marks out of 500: %.1f\n", sum);
    printf("Percentage: %.2f %%\n", (sum / TOTAL) * 100.0);

    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q11.exe
Enter marks of student in 5 subjects:
subject 1: 89
subject 2: 90
subject 3: 92
subject 4: 78
subject 5: 91
Total marks out of 500: 440.0
Percentage: 88.00 %
```

**12. The length and breadth of a rectangle are input through the keyboard. Write a programme to determine (i) radius of a circle and ratio of perimeters of the rectangle and the circle if the areas of the rectangle and the circle are equal (ii) radius of the circle and the ratio of the areas of the rectangle and the circle if the perimeters of the rectangle and the circle are equal.**

**Ans:** CODE—

```
#include <stdio.h>

#include <math.h>

#define PI 3.14

int main()
{
    float length, breadth;

    printf("Enter length and breadth of rectangle: ");
    scanf("%f %f", &length, &breadth);

    // case 1: area of rectangle = area of circle
    printf("\nCASE1: AREA OF RECTANGLE = AREA OF CIRCLE\n");

    float radius = sqrt((length * breadth) / PI);

    // ratio of perimeter of rectangle to circle
    float r1 = ((2 * (length + breadth)) / (2 * PI * radius));
```

```

printf("Radius of circle: %.3f\n", radius);
printf("Ratio of perimeter of rectangle to circumference of circle: %.3f\n", r1);

// case 2: perimeter of rectangle = circumference of circle
printf("\nCASE2: PERIMETER OF RECTANGLE = CIRCUMFERENCE OF CIRCLE\n");

float radius1 = ((2 * (length + breadth)) / (2 * PI));

// ratio of area of rectangle to circle
float r2 = ((length * breadth) / (PI * radius1 * radius1));

printf("Radius of circle: %.3f\n", radius1);
printf("Ratio of area of rectangle to area of circle: %.3f\n", r2);

return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt1>q12.exe
Enter length and breadth of rectangle: 5 8

CASE1: AREA OF RECTANGLE = AREA OF CIRCLE
Radius of circle: 3.569
Ratio of perimeter of rectangle to circumference of circle: 1.160

CASE2: PERIMETER OF RECTANGLE = CIRCUMFERENCE OF CIRCLE
Radius of circle: 4.140
Ratio of area of rectangle to area of circle: 0.743

```

## Assignment 2

1. In a town, the percentage of men is 52. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, write a program to find the total number of illiterate men and women if the population of the town is 80,000.

Ans: CODE –

```
#include <stdio.h>

#define POPULATION 80000

#define MEN_PERCENT 0.52

#define TOTAL_LITERACY_PERCENT 0.48

#define MEN_LITERACY_PERCENT 0.35

int main()
{
    int literate = TOTAL_LITERACY_PERCENT * POPULATION;
    int literate_men = MEN_LITERACY_PERCENT * POPULATION;
    int literate_women = literate - literate_men;
    int total_men = MEN_PERCENT * POPULATION;
    int total_women = POPULATION - total_men;
    int illiterate_women = total_women - literate_women;
    int illiterate_men = total_men - literate_men;

    printf("total illiterate men: %d\n", illiterate_men);
    printf("total illiterate women: %d\n", illiterate_women);

    return 0;
}
```

OUTPUT:

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q13.exe
total illiterate men: 13600
total illiterate women: 28000
```



**2. If a five-digit integer is input through the keyboard, write a program to print a new number by adding one to each of its digits. For example, if the number that is input is 12391 then the output should be displayed as 23402.**

**Ans:** CODE –

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

```
int powerTen(int x)
```

```
{
    int p = 1;
    while (x > 1)
    {
        p *= 10;
        x--;
    }
    return p;
}
```

```
int main()
```

```
{
    int num, newnum = 0, i = 5;
    printf("Enter 5-digit integer: ");
    scanf("%d", &num);
    while (i)
    {
        int digit = num / powerTen(i);
        if (digit == 9)
        {
            newnum = newnum * 10 + 0;
        }
        else
        {
            newnum = newnum * 10 + (num / powerTen(i) + 1);
        }
    }
}
```

```

        num = num % powerTen(i);

        i--;
    }

    printf("new number: %d\n", newnum);

    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q14.exe
Enter 5-digit integer: 12391
new number: 23402

```

**3. Write a program to find the grace marks for a student using switch. The user should enter the class obtained by the student and the number of subjects he has failed in.**

- **If the student gets first class and the number of subjects he failed in is greater than 3, then he does not get any grace. If the number of subjects he failed in is less than or equal to 3 then the grace is of 5 marks per subject.**
- **If the student gets second class and the number of subjects he failed in is greater than 2, then he does not get any grace. If the number of subjects he failed in is less than or equal to 2 then the grace is of 4 marks per subject.**
- **If the student gets third class and the number of subjects he failed in is greater than 1, then he does not get any grace. If the number of subjects he failed in is equal to 1 then the grace is of 5 marks per subject.**

**Ans: CODE –**

```

#include <stdio.h>

int main()
{
    char class_obtained;
    int subjects_failed;

    printf("Enter the class obtained (A/B/C): ");
    scanf(" %c", &class_obtained);

    printf("Enter the number of subjects failed: ");
    scanf("%d", &subjects_failed);

    int grace_marks = 0;

```

```

switch (class_obtained)
{
case 'A':
    if (subjects_failed <= 3)
    {
        grace_marks = 5 * subjects_failed;
    }
    break;

case 'B':
    if (subjects_failed <= 2)
    {
        grace_marks = 4 * subjects_failed;
    }
    break;

case 'C':
    if (subjects_failed == 1)
    {
        grace_marks = 5;
    }
    break;

default:
    printf("Invalid class entered.\n");
    return 1; // Exit with an error code
}

printf("Grace marks: %d\n", grace_marks);

return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q15.exe
Enter the class obtained (A/B/C): B
Enter the number of subjects failed: 2
Grace marks: 8

```

**4. Read a five-letter word into the computer, then encode the word on a letter-by-letter basis by subtracting 30 from the numerical value that is used to represent each letter. Thus, if the ASCII character set is being used, the letter a (which is represented by the value 97) would become a C (represented by the value 67), etc. Write out the encoded version of the word.**

**Ans:**    CODE –

```

#include <stdio.h>

int main()
{
    char word[6], encodedWord[6];
    int i;

    printf("Enter a 5 letter word: ");
    scanf("%s", word);

    for (i = 0; word[i] != '\0'; ++i)
    {
        encodedWord[i] = (char)((int)word[i] - 30);
    }
    encodedWord[i] = '\0';

    printf("Encoded word: %s\n", encodedWord);
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q16.exe
Enter a 5 letter word: Sombra
Encoded word: 5QODT

```

**5. Read into the computer a five-letter word that has been encoded using the scheme described above. Decode the word by reversing the above procedure, then write out the decoded word.**

**Ans:** CODE—

```

#include <stdio.h>

int main()
{

```

```

char word[6], decodedWord[6];

int i;

printf("Enter a 5 letter encoded word: ");
scanf("%[^\\n]c", word);

for (i = 0; word[i] != '\\0'; ++i)
{
    decodedWord[i] = (char)((int)word[i] + 30);
}
decodedWord[i] = '\\0';

printf("Decoded word: %s\\n", decodedWord);

return 0;
}

```

OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q17.exe
Enter a 5 letter encoded word: 5QODT
Decoded word: Sombra

```

**6. Any year is input through the keyboard. Write a program to determine whether the year is a leap year or not. (Hint: Use the % (modulus) operator)**

**Ans:** CODE—

```

#include <stdio.h>

int main()
{
    int year;

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

```

```
if (year % 4 != 0)
{
    printf("%d is not leap year", year);
}
else
{
    if (year % 4 == 0 && year % 100 == 0 && year % 400 == 0)
    {
        printf("%d is a leap year", year);
    }
    else if (year % 4 == 0 && year % 100 == 0 && year % 400 != 0)
    {
        printf("%d is not leap year", year);
    }
    else
    {
        printf("%d is leap year", year);
    }
}
return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q18.exe
Enter year: 2024
2024 is leap year
```

**7. Write a program to check whether a triangle is valid or not, when (i) the three angles of the triangle are entered through the Keyboard (ii) three sides of the triangle are entered through the keyboard.**

**Ans:** CODE –

```
#include <stdio.h>

int main()
{
    float a, b, c, s1, s2, s3;

    // using angles
    printf("Enter three angles of triangle: ");
    scanf("%f %f %f", &a, &b, &c);

    if (a + b + c == 180)
    {
        printf("It forms a triangle\n");
    }
    else
    {
        printf("It does not form a triangle\n");
    }

    // using sides
    printf("Enter three sides of triangle: ");
    scanf("%f %f %f", &s1, &s2, &s3);

    if (s1 + s2 > s3 && s2 + s3 > s1 && s3 + s1 > s2)
    {
        printf("It forms a triangle\n");
    }
    else
    {
```

```

        printf("It does not form triangle\n");
    }
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q19.exe
Enter three angles of triangle: 30 60 90
It forms a triangle
Enter three sides of triangle: 1 2 6
It does not form triangle

```

**8. Given three points (x1, y1), (x2, y2) and (x3, y3), write a program to check if all the three points fall on one straight line.**

**Ans:** CODE—

```

#include <stdio.h>

#define ABS(x) ((x) > 0) ? (x) : (-(x))

int main()
{
    float x1, x2, x3, y1, y2, y3;

    printf("Enter coordinates of 1st point: ");
    scanf("%f %f", &x1, &y1);

    printf("Enter coordinates of 2nd point: ");
    scanf("%f %f", &x2, &y2);

    printf("Enter coordinates of 3rd point: ");
    scanf("%f %f", &x3, &y3);

    if ((0.5 * ABS(x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2))) == 0)
    {

```



```

        printf("The points lie in straight line\n");
    }
    else
    {
        printf("The points do not lie in straight line\n");
    }
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q20.exe
Enter coordinates of 1st point: 1 0
Enter coordinates of 2nd point: -1 0
Enter coordinates of 3rd point: 3 0
The points lie in straight line

```

**9. Given the coordinates (x, y) of a center of a circle and its radius, write a program which will determine whether a point lies inside the circle, on the circle or outside the circle. (Hint: #include <math.h>. Use sqrt( ) and pow( ) functions)**

**Ans:** CODE—

```

#include <stdio.h>

#include <math.h>

int main()
{
    double centerX, centerY, radius, x, y;

    // input center of circle
    printf("Enter the center coordinates (x, y) of the circle: ");
    scanf("%lf %lf", &centerX, &centerY);

    // input radius of circle
    printf("Enter the radius of the circle: ");
    scanf("%lf", &radius);

```

```

// input other point to check

printf("Enter the coordinates (x, y) of the point: ");

scanf("%lf %lf", &x, &y);

// distance between center and given point

double distance = sqrt(pow((x - centerX), 2) + pow((y - centerY), 2));

// comparing distance and radius to check location of point
if (distance < radius)
{
    printf("Point (%.2lf, %.2lf) is inside the circle.\n", x, y);
}
else if (distance == radius)
{
    printf("Point (%.2lf, %.2lf) is on the circle.\n", x, y);
}
else
{
    printf("Point (%.2lf, %.2lf) is outside the circle.\n", x, y);
}

return 0;
}

```

#### OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q21.exe
Enter the center coordinates (x, y) of the circle: 0 0
Enter the radius of the circle: 5
Enter the coordinates (x, y) of the point: 2 3
Point (2.00, 3.00) is inside the circle.

```

**10. Any character is entered through the keyboard, write a program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol.**

**Ans:** CODE—

```
#include <stdio.h>

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

    if (ascii >= 65 && ascii <= 90)
    {
        printf("Upper case character\n");
    }
    else if (ascii >= 97 && ascii <= 122)
    {
        printf("Lower case character\n");
    }
    else if (ascii >= 48 && ascii <= 57)
    {
        printf("Digit character\n");
    }
    else
    {
        printf("Other special character\n");
    }
    return 0;
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q22.exe
Enter any character: #
Other special character
```

**11. Given as input an integer number of seconds, write a program to print as output the equivalent time in hours, minutes, and seconds. Recommended output format is something like 7322 seconds is equivalent to 2 hours 2 minutes 2 seconds.**

**Ans:** CODE —

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

```
int main()
```

```
{
```

```
    int seconds;
```

```
    printf("Enter time in seconds: ");
```

```
    scanf("%d", &seconds);
```

```
    int h = seconds / 3600;
```

```
    int m = (seconds - (h * 3600)) / 60;
```

```
    int s = (seconds - (h * 3600) - (m * 60));
```

```
    printf("Time: %d hours %d minutes %d seconds\n", h, m, s);
```

```
    return 0;
```

```
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q23.exe
Enter time in seconds: 7322
Time: 2 hours 2 minutes 2 seconds
```

**12. Write a program which accepts two number X, Y and creates a third number Z by appending Y after X. Example: if X=12 and Y=345 then Z=12345.**

**Ans:** CODE—

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

```
int countLength(int n)
```

```
{
```

```
    int c = 0;
```

```
    while (n)
```

```
    {
```

```
        c += 1, n /= 10;
```

```
    }
```

```
    return c;
```

```
}
```

```
int powerTen(int x)
```

```
{
```

```
    int p = 1;
```

```
    while (x >= 1)
```

```
    {
```

```
        p *= 10;
```

```
        x--;
```

```
    }
```

```
    return p;
```

```
}
```

```
int main()
```

```
{
```

```
    int X, Y;
```

```
    printf("Enter two numbers:\n");
```

```
    printf("X = ");
```

```

scanf("%d", &X);

printf("Y = ");

scanf("%d", &Y);

// We form Z by concatenating X and Y
// example: X = 12, Y = 345, then Z = 12345

int Z = (X * powerTen(countLength(Y))) + Y;

printf("X = %d, Y = %d\n", X, Y);

printf("Z = %d\n", Z);

return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q24.exe
Enter two numbers:
X = 12
Y = 345
X = 12, Y = 345
Z = 12345

```

**13. A certain grade of steel is graded according to the following conditions:**

**Hardness must be greater than 50**

**Carbon content must be less than 0.7**

**Tensile strength must be greater than 5600**

**The grades are as follows:**

**Grade is 10 if all three conditions are met**

**Grade is 9 if conditions (i) and (ii) are met**

**Grade is 8 if conditions (ii) and (iii) are met**

**Grade is 7 if conditions (i) and (iii) are met**

**Grade is 6 if only one condition is met**

**Grade is 5 if none of the conditions are met**

**Write a program, which will require the user to give values of hardness, carbon content and tensile strength of the steel under consideration and output the grade of the steel.**

**Ans:    CODE –**

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

```
int main()
```

```
{
```

```
    // Input values
```

```
    double hardness, carbonContent, tensileStrength;
```

```
    printf("Enter hardness, carbon content, and tensile strength of the steel: ");
```

```
    scanf("%lf %lf %lf", &hardness, &carbonContent, &tensileStrength);
```

```
    // Check conditions and determine grade
```

```
    int grade;
```

```
    if (hardness > 50 && carbonContent < 0.7 && tensileStrength > 5600)
```

```
    {
```

```
        grade = 10;
```

```
    }
```

```
    else if (hardness > 50 && carbonContent < 0.7)
```

```
    {
```

```
        grade = 9;
```

```
    }
```

```
    else if (carbonContent < 0.7 && tensileStrength > 5600)
```

```
    {
```

```
        grade = 8;
```

```
    }
```

```

else if (hardness > 50 && tensileStrength > 5600)
{
    grade = 7;
}
else if (hardness > 50 || carbonContent < 0.7 || tensileStrength > 5600)
{
    grade = 6;
}
else
{
    grade = 5;
}

// Output the grade
printf("Grade of the steel: %d\n", grade);

return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt2>q25.exe
Enter hardness, carbon content, and tensile strength of the steel: 52 0.6 6000
Grade of the steel: 10

```



## Assignment 3

1. Write a C program which prints all integers divisible by n between 1 and 100 where value of n is provided by the user.

Ans: CODE—

```
#include <stdio.h>

int main()
{
    int n;

    // Input the value of n from the user
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Print integers divisible by n between 1 and 100
    printf("Integers divisible by %d between 1 and 100:\n", n);
    for (int i = 1; i <= 100; i++)
    {
        if (i % n == 0)
        {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

OUTPUT—

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q1.exe
Enter the value of n: 7
Integers divisible by 7 between 1 and 100:
7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98,
```

**2. Write a C program to find out sum of digits of a given number.**

**Ans:** CODE—

```
#include <stdio.h>

int main()
{
    int number, originalNumber, sum = 0;

    // Input the number from the user
    printf("Enter a number: ");
    scanf("%d", &number);

    // Save the original number for reference
    originalNumber = number;

    while (number > 0)
    {
        int digit = number % 10;
        sum += digit;
        number /= 10;
    }
    printf("Sum of digits of %d is: %d\n", originalNumber, sum);
    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q2.exe
Enter a number: 21314
Sum of digits of 21314 is: 11
```

**3. Write a C program to find the reverse of a given number.**

**Ans:** CODE—

```
#include <stdio.h>

int main()
{
    int number, reversedNumber = 0;

    // Input the number from the user
    printf("Enter a number: ");
    scanf("%d", &number);

    // Calculate the reverse of the number
    while (number > 0)
    {
        int digit = number % 10;
        reversedNumber = reversedNumber * 10 + digit;
        number /= 10;
    }
    printf("Reverse of the given number is: %d\n", reversedNumber);
    return 0;
}
```

OUTPUT —

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q3.exe
Enter a number: 13214
Reverse of the given number is: 41231
```

**4. Write a C program to find out sum of the following series.  $S=1+2+3+4+ \dots +n$**

**Ans- CODE—**

```
#include <stdio.h>

int main()
{
    int n, s = 0;
    printf("Enter n: ");
    scanf("%d", &n);

    for (int i = 1; i <= n; ++i)
    {
        s += i;
    }
    printf("Sum: %d", s);
    return 0;
}
```

**OUTPUT —**

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q4.exe
Enter n: 12
Sum: 78
```

**5. Write a C program to find out sum of the following series.  $S=1.2+2.3+3.4+4.5+ \dots +n.(n+1)$**

**Ans: CODE—**

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

```
int main()
{
```

```
int n, s = 0;

printf("Enter n: ");

scanf("%d", &n);

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

printf("Sum: %d", s);

return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q5.exe
Enter n: 5
Sum: 70
```

**6. Write a C program to find out factorial of a given number.**

**Ans:** CODE—

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

```
int factorial(int n)
{
    int f = 1, i = 1;
    while (i <= n)
    {
        f *= i;
        i++;
    }
}
```

```

    return f;
}

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

    if (n < 0)
    {
        printf("Factorial of negative number does not exist\n");
    }
    else
    {
        printf("Factorial of %d is %d\n", n, factorial(n));
    }
    return 0;
}

```

OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q6.exe
Enter number: 5
Factorial of 5 is 120

```

**7. C program to find out sum of the following series.  $S=1!+2!+3!+4!+ \dots +n!$**

**Ans:** CODE—

```

#include <stdio.h>

int factorial(int n)

```

```

{
    int f = 1, i = 1;
    while (i <= n)
    {
        f *= i;
        i++;
    }
    return f;
}

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

    for (int i = 1; i <= n; ++i)
    {
        s += factorial(i);
    }

    printf("Sum: %d\n", s);
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q7.exe
Enter number: 5
Sum: 153

```

**8. Write a C program to find out sum of the following series.  $S = 1@ + 2@ + 3@ + 4@ + \dots + n@$**

**where,  $n@$  is the sum of all factors of  $n$ . Example:  $6@ = 1+2+3+6 = 12$**

**Ans: CODE—**

```
#include <stdio.h>

int sumFactors(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; ++i)
    {
        // checking for factor
        if (n % i == 0)
        {
            sum += i;
        }
    }
    return sum;
}

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

    for (int i = 1; i <= n; ++i)
    {
        s += sumFactors(i);
    }

    printf("Sum: %d\n", s);

    return 0;
}
```

**OUTPUT –**

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q8.exe
Enter number: 6
Sum: 33
```

**9. Write a C program which prints all prime numbers between 1 and 100.**

**Ans: CODE—**



```
#include <stdio.h>

#include <math.h>

int checkPrime(int n)
{
    // by-default c = 2 because 1 and number itself are already factors
    int c = 2, flag = 0;

    for (int i = 2; i <= sqrt(n); ++i)
    {
        if (n % i == 0)
        {
            c += 1;
        }

        if (c > 2)
        {
            // composite. because number of factors greater than 2
            flag = 1;
            break;
        }
    }

    // prime
    if (flag == 0)
    {
        return 1;
    }

    // composite
    else
```

```

    {
        return 0;
    }
}

int main()
{
    printf("The prime numbers between 1 and 100:\n");
    for (int i = 1; i <= 100; ++i)
    {
        if (checkPrime(i) && i != 1)
        {
            printf("%d ", i);
        }
    }
    return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q9.exe
The prime numbers between 1 and 100:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

```

**10. Write a program which accepts a number n and prints all prime factors of n.**

**Ans:** CODE—

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

```
#include <math.h>
```

```
int checkPrime(int n)
```

```
{  
    // by-default c = 2 because 1 and number itself are already factors  
    int c = 2, flag = 0;  
  
    for (int i = 2; i <= sqrt(n); ++i)  
    {  
        if (n % i == 0)  
        {  
            c += 1;  
        }  
  
        if (c > 2)  
        {  
            // composite. because number of factors greater than 2  
            flag = 1;  
            break;  
        }  
    }  
  
    // prime  
    if (flag == 0)  
    {  
        return 1;  
    }  
  
    // composite  
    else  
    {  
        return 0;  
    }  
}
```

```

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

    printf("The prime factors of %d:\n", n);

    for (int i = 2; i <= n; ++i)
    {
        if (n % i == 0 && checkPrime(i))
        {
            printf("%d ", i);
        }
    }
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q10.exe
Enter number: 36
The prime factors of 36:
2 3

```

**11. Write a program to generate all combinations of digit 1, 2 and 3 using for loop.**

**Ans:** CODE–

```

#include <stdio.h>

```

```

int main()
{

```

```

printf("\nAll possible combinations of digits 1, 2, 3:\n");
for (int i = 1; i <= 3; i++)
{
    for (int j = 1; j <= 3; j++)
    {
        for (int k = 1; k <= 3; k++)
        {
            printf("%d%d%d, ", i, j, k);
        }
    }
}
printf("\n");
return 0;
}

```

#### OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q11.exe
```

```
All possible combinations of digits 1, 2, 3:
```

```
111, 112, 113, 121, 122, 123, 131, 132, 133, 211, 212, 213, 221, 222, 223, 231, 232, 233, 311, 312, 313, 321, 322, 323, 331, 332, 333,
```

**12. Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form.**

**29 \* 1 = 29**

**29 \* 2 = 58....**

**Ans: CODE –**

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

```
int main()
```

```
{
```

```
    int n;
```

```

printf("Enter number whose table is to be printed: ");
scanf("%d", &n);

printf("MULTIPLICATION TABLE OF %d:\n", n);
for (int i = 1; i <= 10; ++i)
{
    printf("%d x %d = %d\n", n, i, n * i);
}

return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q12.exe
Enter number whose table is to be printed: 29
MULTIPLICATION TABLE OF 29:
29 x 1 = 29
29 x 2 = 58
29 x 3 = 87
29 x 4 = 116
29 x 5 = 145
29 x 6 = 174
29 x 7 = 203
29 x 8 = 232
29 x 9 = 261
29 x 10 = 290

```

**13. Write a menu driven program to accept a number in any Number System [from Binary, Decimal, Octal, and Hex] and convert and display the same in any other amongst these.**

**Ans:** CODE –

```

#include <stdio.h>
#include <string.h>
int ctoi(char c)

```

```
{  
    return c - '0';  
}
```

char itoc(int i)

```
{  
    return i + '0';  
}
```

int decimalToBase(int n, int b)

```
{  
    // converts decimal number to any base (except hexadecimal)  
    int s = 0;  
    if (n == 0)  
    {  
        return s;  
    }  
    else  
    {  
        s = (n % b) + (10 * decimalToBase(n / b, b));  
    }  
}
```

int getPower(int x, int y)

```
{  
    // returns x ^ y  
    int p = 1;  
    while (y)  
    {  
        p *= x;  
        y--;  
    }
```

```

    }

    return p;
}

int baseToDecimal(int n, int b)
{
    // converts any base (except hexadecimal) to decimal
    int s = 0, i = 0;
    while (n)
    {
        int rem = n % 10;
        s += rem * getPower(b, i);
        i++;
        n /= 10;
    }

    return s;
}

int hexToDecimal(char h[])
{
    int l = strlen(h), i = l - 1, digit, j = 0, s = 0;

    while (i >= 0)
    {
        switch (h[i])
        {
            case 'A':
            case 'a':
                digit = 10;
                break;

```



```
case 'B':
```

```
case 'b':
```

```
    digit = 11;
```

```
    break;
```

```
case 'C':
```

```
case 'c':
```

```
    digit = 12;
```

```
    break;
```

```
case 'D':
```

```
case 'd':
```

```
    digit = 13;
```

```
    break;
```

```
case 'E':
```

```
case 'e':
```

```
    digit = 14;
```

```
    break;
```

```
case 'F':
```

```
case 'f':
```

```
    digit = 15;
```

```
    break;
```

```
default:
```

```
    digit = atoi(h[i]);
```

```
    break;
```

```
}
```

```
s += digit * getPower(16, j);
```

```
        i--;

        j++;
    }

    return s;
}

void decToHex(int n)
{
    char hex[15];
    int i = 0;
    while (n)
    {
        int rem = n % 16;
        switch (rem)
        {
            case 10:
                hex[i++] = 'A';
                break;

            case 11:
                hex[i++] = 'B';
                break;

            case 12:
                hex[i++] = 'C';
                break;

            case 13:
                hex[i++] = 'D';
                break;
```

case 14:

hex[i++] = 'E';

break;

case 15:

hex[i++] = 'F';

break;

default:

hex[i++] = itoc(rem);

}

n /= 16;

}

// if decimal number = 0

if (i == 0)

{

hex[i++] = '0';

}

hex[i] = '\0';

// reversing the hex array to get correct order

for (int j = 0; j < i / 2; ++j)

{

char temp = hex[j];

hex[j] = hex[i - j - 1];

hex[i - j - 1] = temp;

}

printf("%s\n", hex);

}

```
int main()
{
    int ib, ob, n;
    char hex[15];

    printf("Enter input base: ");
    scanf("%d", &ib);
    printf("Enter output base: ");
    scanf("%d", &ob);

    if (ib != 16)
    {
        printf("Enter number in input base: ");
        scanf("%d", &n);

        if (ob != 16)
        {
            printf("Number in output base: %d\n", decimalToBase(baseToDecimal(n, ib), ob));
        }
        else
        {
            printf("Number in output base: ");
            decToHex(baseToDecimal(n, ib));
        }
    }

    else
    {
        printf("Enter number in input base: ");
        scanf("%s", hex);
    }
}
```

```

    printf("Number in output base: %d\n", decimalToBase(hexToDecimal(hex), ob));
}
return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q13.exe
Enter input base: 10
Enter output base: 16
Enter number in input base: 2142
Number in output base: 85E

```

**14. Write a menu-driven program for displaying log-series results for an input number for different log bases.**

**Ans:** CODE –

```

#include <stdio.h>
#include <math.h>

int main(void)
{
    float a, b;

    printf("Enter value of a and b for log_b(a): ");
    scanf("%f %f", &a, &b);

    printf("Log of %.2f to the base %.2f = log_%.2f(%.2f) = %.2f\n", a, b, b, a, log(a) / log(b));
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q14.exe
Enter value of a and b for log_b(a): 3 8
Log of 3.00 to the base 8.00 = log_8.00(3.00) = 0.53

```

**15. Write a program named SINE to find the sine of an angle. The angle and its unit (degree, radian, or grade) should be provided as command line arguments. For the units, short forms as d/D (for degree), r/R (for radian) or g/G (for grade) may be used. The program should use the series:**

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \dots$$

**for evaluation. Take care of negative angles and angles in all the quadrants.**

**Ans: CODE –**

```
#include <stdio.h>
#include <math.h>
#define PI 3.14
#define EPSILON 0.00001

double factorial(int x)
{
    double f = 1;
    int i = 1;
    while (i <= x)
    {
        f *= i;
        i++;
    }
    return f;
}

int main(void)
{
    double x, s = 0.0, sign = 1.0, term;
    int i = 0;

    printf("Enter x: ");
    scanf("%lf", &x);
```

```

printf("sin(%.2lf) = ", x);

// bringing x in [-pi, pi] interval to increase the convergence rate of the Taylor series
while (x > PI)
{
    x -= 2 * PI;
}

while (x < -PI)
{
    x += 2 * PI;
}

// actual value for testing
// printf("%.8lf\n", sin(x));

do
{
    term = sign * (pow(x, 2 * i + 1) / factorial(2 * i + 1));
    s += term;
    sign *= -1, i++;
} while (fabs(term) > EPSILON);

printf("%.8lf\n", s);
return 0;
}

```

#### OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q15.exe
Enter x: -32.23
sin(-32.23) = -0.73793137

```

**16. Write a C program which accepts 10 integers and displays the counts of positives, negatives and zeros entered.**

**Ans:** CODE –

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

```
int main()
```

```
{
```

```
    int i, p = 0, n = 0, z = 0;
```

```
    printf("Enter 10 integers one by one:\n");
```

```
    for (int j = 0; j < 10; ++j)
```

```
    {
```

```
        scanf("%d", &i);
```

```
        if (i > 0)
```

```
        {
```

```
            p += 1;
```

```
        }
```

```
        else if (i == 0)
```

```
        {
```

```
            z += 1;
```

```
        }
```

```
        else
```

```
        {
```

```
            n += 1;
```

```
        }
```

```
    }
```

```
    printf("\nNumber of positive integers: %d\n", p);
```

```
    printf("Number of zeroes: %d\n", z);
```



```
printf("Number of negative integers: %d\n", n);  
return 0;  
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q16.exe  
Enter 10 integers one by one:  
4  
0  
-5  
9  
-4  
2  
10  
-1  
0  
7  
  
Number of positive integers: 5  
Number of zeroes: 2  
Number of negative integers: 3
```

**17. Write a C program to print the first n numbers of the Fibonacci sequence. The Fibonacci sequence is constructed by adding the last two numbers of the sequence so far to get the next number in the sequence. The first and second numbers of the sequence are defined as 0 and 1. We get:**

**0, 1, 1, 2, 3, 5, 8, 13, 21...**

**Ans: CODE –**

```
#include <stdio.h>  
  
int main(void)  
{  
    int first = 0, second = 1, n;  
    printf("Enter n: ");  
    scanf("%d", &n);  
  
    printf("%d, %d", first, second);  
  
    while (n > 2)  
    {
```

```

    int third = first + second;
    printf(", %d", third);
    first = second;
    second = third;
    --n;
}
return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q17.exe
Enter n: 8
0, 1, 1, 2, 3, 5, 8, 13

```

**18. Write a program to print out all Armstrong numbers between 1 and 500. If the sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example,  $153 = (1^3) + (5^3) + (3^3)$ .**

**Ans:** CODE –

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

```
int checkArmstrong(int x)
```

```

{
    int x1 = x, sum = 0;
    while (x > 0)
    {
        sum += (x % 10) * (x % 10) * (x % 10);
        x /= 10;
    }

```

```
if (sum == x1)
```

```

{
    return 1;
}

```

```
else
```

```

    {
        return 0;
    }
}

int main(void)
{
    printf("The armstrong numbers between 1 and 500 are: ");
    for (int i = 1; i <= 500; ++i)
    {
        if (checkArmstrong(i))
        {
            printf("%d ", i);
        }
    }
    printf("\n");
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q18.exe
The armstrong numbers between 1 and 500 are: 1 153 370 371 407

```

**19. Write a C program which prints the first 10 happy numbers. If you iterate the process (assume maximum 100 iterations) of summing the squares of the decimal digits of a number and if the process terminates in 1, then the original number is called a Happy number. For example, 7 is a happy number as  $7 \rightarrow 49 \rightarrow 97 \rightarrow 130 \rightarrow 10 \rightarrow 1$ .**

**Ans:** CODE –

```

// check happy number

#include <stdio.h>

int sumSquaresDigits(int x)
{
    int s = 0;

```

```

while (x > 0)
{
    s += (x % 10) * (x % 10);
    x /= 10;
}

return s;
}

int checkHappy(int x)
{
    int iterations = 100;

    while (iterations > 0)
    {
        x = sumSquaresDigits(x);

        if (x == 1)
        {
            return 1; // Happy number
        }

        iterations--;
    }

    return 0; // Not a happy number within 100 iterations
}

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

    if (checkHappy(n))
    {
        printf("%d is a happy number", n);
    }
    else
    {
        printf("%d is not a happy number", n);
    }

    return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q19.exe
Enter number: 44
44 is a happy number

```

**20. An important property of square numbers: If a natural number is a square number, then it must be the sum of Successive Odd Numbers starting from 1. For example: Perfect Square Sum of Odd Numbers:**

$$4 = 1 + 3$$

$$9 = 1 + 3 + 5$$

$$16 = 1 + 3 + 5 + 7$$

$$25 = 1 + 3 + 5 + 7 + 9$$

$$36 = 1 + 3 + 5 + 7 + 9 + 11$$

$$49 = 1 + 3 + 5 + 7 + 9 + 11 + 13$$

$$64 = 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15$$

**Now using this property, find the square root of any perfect square.**

**Ans:**    CODE –

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

```
int main(void)
```

```
{
```

```
    int ps, sqrt = 0, i = 1, sum = 0;
```

```
    printf("Enter a perfect square number: ");
```

```
    scanf("%d", &ps);
```

```
    while (sum < ps)
```

```
    {
```

```
        sum += i;
```

```
        i += 2, sqrt += 1;
```

```
    }
```

```
    printf("%d\n", sum);
```

```
    printf("The square root of %d is %d\n", ps, sqrt);
```

```
    return 0;
```

```
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q20.exe
Enter a perfect square number: 49
49
The square root of 49 is 7
```

**21. Write a program which reads a positive integer value, and compute the following sequence:**

**if the number is even, halve it; if it is odd, multiply by 3 and add 1.**

**Repeat this process until the value is 1, printing out each intermediate value.**

**Finally, the program should print how many of these operations were performed.**

**Typical output might be:**

```
Initial value is 3
Next value is 10
Next value is 5
Next value is 16
Next value is 8
Next value is 4
Next value is 2
Next value is 1
Number of operations is 7
```

**Ans: CODE –**

```
#include <stdio.h>

int main(void)
{
    int n, ops = 0;
    printf("Enter initial value: ");
    scanf("%d", &n);
    printf("Initial value is %d\n\n", n);
    while (n != 1)
    {
```

```

    if (n % 2 == 0)
    {
        n = n / 2;
    }
    else
    {
        n = 3 * n + 1;
    }
    printf("Next value is %d\n", n);
    ops += 1;
}
printf("Next value is %d\n", n);
printf("Number of operations is %d\n", ops);
}

```

#### OUTPUT:

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q21.exe
Enter initial value: 3
Initial value is 3

Next value is 10
Next value is 5
Next value is 16
Next value is 8
Next value is 4
Next value is 2
Next value is 1
Next value is 1
Number of operations is 7

```

**22. Write a program to print all the ASCII values and their equivalent characters using a while loop. The ASCII values vary from 0 to 255.**

**Ans:** CODE –

```

#include <stdio.h>

```

```

int main(void)

```

```

{

```

```
printf("From ascii 0 to 31 and from 127 to 255, the characters are not printable.\n");
```

```
printf("\nFrom 32 to 126:\n\n");
```

```
printf("ascii\t\tcharacter\n");
```

```
for (int i = 32; i <= 126; ++i)
```

```
{
```

```
    printf("%d\t\t%c\n", i, (char)i);
```

```
}
```

```
return 0;
```

```
}
```

OUTPUT – (**note:** ascii values from 0 to 32, and 127 to 255 are not printable)

ascii	character	81	Q
32		82	R
33	!	83	S
34	"	84	T
35	#	85	U
36	\$	86	V
37	%	87	W
38	&	88	X
39	'	89	Y
40	(	90	Z
41	)	91	[
42	*	92	\
43	+	93	]
44	,	94	^
45	-	95	_
46	.	96	`
47	/	97	a
48	0	98	b
49	1	99	c
50	2	100	d
51	3	101	e
52	4	102	f
53	5	103	g
54	6	104	h
55	7	105	i
56	8	106	j
57	9	107	k
58	:	108	l
59	;	109	m
60	<	110	n
61	>	111	o
62	?@	112	p
63	?	113	q
64	@	114	r
65	A	115	s
66	B	116	t
67	C	117	u
68	D	118	v
69	E	119	w
70	F	120	x
71	G	121	y
72	H	122	z
73	I	123	{
74	J	124	
75	K	125	}
76	L	126	~

**24. Write a program to find the octal equivalent of the entered integer.**

**Ans:** CODE –

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

```
int octal(int n)
```

```
{
```

```
    int oct = 0;
```



```

if (n == 0)
{
    return oct;
}
else
{
    oct = (n % 8) + (10 * octal(n / 8));
}
}

```

```

int main()
{
    int decimal;
    printf("Enter the decimal: ");
    scanf("%d", &decimal);
    int oct = octal(decimal);
    printf("Octal: %d\n", oct);
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q23.exe
Enter the decimal: 100
Octal: 144

```

**24. Write a C program that prints the following patterns for the input of n=4. The value of n is input by the user.**

**Ans:**    CODE –

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

```
// Pattern 1
```

```
void pattern1(int n)
```

```
{  
    for (int row = 1; row <= n; ++row)  
    {  
        for (int col = 1; col <= row; ++col)  
        {  
            printf("%d ", col);  
        }  
  
        printf("\n");  
    }  
}
```

```
// Pattern 2
```

```
void pattern2(int n)
```

```
{  
    int space = 2 * (n - 1);  
  
    for (int row = 1; row <= n; ++row)  
    {  
        for (int k = 0; k < space; ++k)  
        {  
            printf(" ");  
        }  
        for (int col = 1; col <= row; ++col)  
        {  
            printf("%d ", col);  
        }  
  
        space -= 2;  
    }  
}
```

```
        printf("\n");
    }
}

// Pattern 3
void pattern3(int n)
{
    int space = 2 * (n - 1);

    for (int row = 1; row <= n; ++row)
    {
        for (int k = 0; k < space; ++k)
        {
            printf(" ");
        }

        for (int col = 1; col <= row; ++col)
        {
            printf("%d ", col);
        }

        for (int col = row - 1; col >= 1; --col)
        {
            printf("%d ", col);
        }

        space -= 2;
        printf("\n");
    }
}
```

```
void pattern4(int n)
{
    int space = 2 * n - 2;

    for (int i = 1; i <= 2 * n - 1; ++i)
    {
        // printing the spaces
        for (int m = 0; m < space; ++m)
        {
            printf(" ");
        }

        // printing the numbers
        if (i % 2 != 0)
        {
            int j;

            if (i == 1)
            {
                printf("1");
            }
            else
            {
                for (j = 1; j < i; ++j)
                {
                    if (j > (i / 2) + 1)
                    {
                        break;
                    }
                    printf("%d ", j);
                }
            }
        }
    }
}
```

```
    for (int k = j - 2; k >= 1; --k)
    {
        printf("%d ", k);
    }
}
```

else

```
{
    int j;
    if (i == 2)
    {
        printf("1 1");
    }
    else
    {
        for (j = 1; j < 2 * i; ++j)
        {
            if (j > i / 2)
            {
                break;
            }
            printf("%d ", j);
        }
        // printf("j = %d", j);
    }
}
```

```
for (int k = j - 1; k >= 1; --k)
{
    printf("%d ", k);
}
```

```

        }
    }
    printf("\n");
    space -= 1;
}
}

```

```

void pattern5(int n)
{
    int k = n;

    for (int i = 1; i <= n - 1; ++i)
    {
        for (int j = k; j >= i - 1; --j)
        {
            printf("%d ", j);
        }
        printf("\n");
        --k;
    }
}

```

```

void pattern6(int n)
{
    int space = n - 1, plus = 1;
    for (int row = 1; row <= 2 * n - 1; ++row)
    {

        if (row <= n)
        {
            // printf("row = %d and space = %d and plus = %d\n", row, space, plus);

```

```
for (int i = 0; i < space; ++i)
```

```
{
```

```
    printf(" ");
```

```
}
```

```
for (int i = 0; i < plus; ++i)
```

```
{
```

```
    printf("+");
```

```
}
```

```
space -= 1, plus += 2;
```

```
}
```

```
else
```

```
{
```

```
    if (space == -1 && plus == 2 * n + 1)
```

```
    {
```

```
        space = 1, plus -= 4;
```

```
    }
```

```
for (int i = 0; i < space; ++i)
```

```
{
```

```
    printf(" ");
```

```
}
```

```
for (int i = 0; i < plus; ++i)
```

```
{
```

```
    printf("+");
```

```
}
```

```
space += 1, plus -= 2;
```

```

    }

    printf("\n");
}
}

void pattern7(int n)
{
    int space = 2 * (n - 1);

    for (int row = 1; row <= n; ++row)
    {
        int col;
        for (int k = 0; k < space; ++k)
        {
            printf(" ");
        }

        for (col = 1; col <= row; ++col)
        {
            printf("%d ", col);
        }

        if (row != 1)
        {
            // why col1 = col - 2
            // because for the previous loop, suppose in row3, col = 4 after previous loop termination
            // so we need to decrement col by 2 (4 - 2 = 2) to start printing col again from 2->1
            for (int col1 = col - 2; col1 >= 1; --col1)
            {
                printf("%d ", col1);
            }
        }
    }
}

```



```

        }
    }

    space -= 2;
    printf("\n");
}
}

void pattern8(int n)
{
    int midspace = 1;

    // topmost row with no midspace
    for (int col = 0; col < 2 * n + 1; ++col)
    {
        printf("*");
    }
    printf("\n");

    // top half
    for (int row = 0; row < n; ++row)
    {
        for (int col = n - row; col >= 1; --col)
        {
            printf("*");
        }

        for (int col = 0; col < midspace; ++col)
        {
            printf(" ");
        }

        for (int col = n - row; col >= 1; --col)

```

```
{
    printf("*");
}

midspace += 2;
printf("\n");
}

// bottomhalf
midspace -= 4;
int j = 0;

for (int row = n; row < 2 * n - 1; ++row)
{
    for (int col = 0; col < 2 + j; ++col)
    {
        printf("*");
    }

    for (int col = 0; col < midspace; ++col)
    {
        printf(" ");
    }

    for (int col = 0; col < 2 + j; ++col)
    {
        printf("*");
    }

    j += 1;
    midspace -= 2;
    printf("\n");
}
```

```
}

// bottommost row with no spaces
for (int col = 0; col < 2 * n + 1; ++col)
{
    printf("*");
}
printf("\n");
}
```

```
void pattern9(int n)
{
    int totalRows = n * 2 - 1;
    int i, j;

    for (i = 1; i <= totalRows; i++)
    {
        for (j = 1; j <= totalRows; j++)
        {
            if (i <= n)
            {
                if (j == n - i + 1 || j == n + i - 1)
                {
                    printf("*");
                }
                else
                {
                    printf(" ");
                }
            }
            else
            {
```

```

        {
            if (j == i - n + 1 || j == totalRows - i + n)
            {
                printf("*");
            }
            else
            {
                printf(" ");
            }
        }
    }
    printf("\n");
}

```

```
int main(void)
```

```

{
    int n;

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

    printf("\nPattern 1:\n\n");
    pattern1(n);

    printf("\n\nPattern 2:\n\n");
    pattern2(n);

    printf("\n\nPattern 3:\n\n");
    pattern3(n);

    printf("\n\nPattern 4:\n\n");
    pattern4(n);

    printf("\n\nPattern 5:\n\n");
    pattern5(n);

    printf("\n\nPattern 6:\n\n");
}

```

```

pattern6(n);

printf("\n\nPattern 7:\n\n");

pattern7(n);

printf("\n\nPattern 8:\n\n");

pattern8(n);

printf("\n\nPattern 9:\n\n");

pattern9(n);

return 0;

}

```

OUTPUT –

<pre> D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3&gt;q24.exe Enter the value of n: 4  Pattern 1: 1 1 2 1 2 3 1 2 3 4  Pattern 2:     1   1 2 1 2 3 1 2 3 4  Pattern 3:     1   1 2 1 1 2 3 2 1 1 2 3 4 3 2 1  Pattern 4:     1   1 1 1 2 1 1 2 2 1 1 2 3 2 1 1 2 3 3 2 1 1 2 3 4 3 2 1  Pattern 5: 4 3 2 1 0 3 2 1 2 </pre>	<pre> Pattern 6: + +++ +++++ +++++++ +++++ +++ +  Pattern 7:     1   1 2 1 1 2 3 2 1 1 2 3 4 3 2 1  Pattern 8: ***** ***** ***** *** ** * * * * * * * *****  Pattern 9: * * * * * * * * * * * * </pre>
---	--

**25. Write a program to add first seven terms of the following series using a for loop:  $1/1! + 2/2! + 3/3! + \dots$**

**Ans:** CODE—

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

```

float factorial(int x)
{
    int i = 1, f = 1.0;
    while (i <= x)
    {
        f *= i;
        ++i;
    }

    return f;
}

int main(void)
{
    float sum = 0;
    for (int i = 1; i <= 7; ++i)
    {
        sum += (i / factorial(i));
    }

    printf("The sum of 1/1! + 2/2! + ... + 7/7! is: \n%.3f\n", sum);
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt3>q25.exe
The sum of 1/1! + 2/2! + ... + 7/7! is:
2.718

```

P.T.O

# Assignment 4

1. Write a program in C to reverse the contents of the elements of an integer array.

Ans: CODE –

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

```
void print_array(int a[], int n)
```

```
{
    for (int i = 0; i < n; ++i)
    {
        printf("%d ", a[i]);
    }
}
```

```
void reverse_array(int a[], int n)
```

```
{
    int rev[n];
    for (int i = 0; i < n; ++i)
    {
        rev[i] = a[n - i - 1];
    }
}
```

```
printf("Reversed array: ");
```

```
print_array(rev, n);
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("Enter size of array: ");
```

```
scanf("%d", &n);

int arr[n];

// input array
printf("Enter %d array elements:\n", n);
for (int i = 0; i < n; ++i)
{
    scanf("%d", &arr[i]);
}

printf("Array entered: ");
print_array(arr, n);

printf("\n");
reverse_array(arr, n);

return 0;
}
```

#### OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q1.exe
Enter size of array: 6
Enter 6 array elements:
1 8 3 9 7 2
Array entered: 1 8 3 9 7 2
Reversed array: 2 7 9 3 8 1
```



**2. Write a program in C to read n number of values in an array. After that, count the total number of duplicate elements in that array. Then copy the elements except the duplicate elements of that array into another array and display this array in reverse order.**

**Ans:** CODE—

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

```
int main()
```

```
{
```

```
    int n, dup = 0;
```

```
    printf("Enter size of array: ");
```

```
    scanf("%d", &n);
```

```
    int arr[n], unique[n], c = 0;
```

```
    printf("Enter %d array elements: ", n);
```

```
    for (int i = 0; i < n; ++i)
```

```
    {
```

```
        scanf("%d", &arr[i]);
```

```
    }
```

```
    for (int i = 0; i < n; ++i)
```

```
    {
```

```
        int is_unique = 1;
```

```
        for (int j = 0; j < c; ++j)
```

```
        {
```

```
            if (arr[i] == unique[j])
```

```
            {
```

```
                dup += 1;
```

```
                is_unique = 0;
```

```
            }
```

```
        }
```

```

        if (is_unique)
        {
            unique[c++] = arr[i];
        }
    }

    printf("Number of duplicate elements: %d\n", dup);

    // printing unique element array
    printf("Unique array: ");
    for (int i = 0; i < c; ++i)
    {
        printf("%d ", unique[i]);
    }
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q2.exe
Enter size of array: 7
Enter 7 array elements: 1 1 6 8 8 2 3
Number of duplicate elements: 2
Unique array: 1 6 8 2 3

```

**3. Write a menu-driven program for accepting values in two square matrix of 3x3 dimension and generate their sum, difference and product.**

**Ans:** CODE –

```

#include <stdio.h>

#include <stdlib.h>

```

```
float **input_matrix()
{
    printf("\nEnter matrix :\n");

    float **mat;
    mat = (float **)malloc(3 * sizeof(float *));
    for (int i = 0; i < 3; ++i)
    {
        *(mat + i) = (float *)malloc(3 * sizeof(float));
        for (int j = 0; j < 3; ++j)
        {
            scanf("%f", &mat[i][j]);
        }
    }
    return mat;
}
```

```
float **createNullMatrix()
{
    float **mat;
    mat = (float **)calloc(3, sizeof(float *));
    for (int i = 0; i < 3; ++i)
    {
        *(mat + i) = (float *)calloc(3, sizeof(float));
    }
    return mat;
}
```

```
void print_matrix(float **m)
{
    for (int i = 0; i < 3; ++i)
```

```
{  
    for (int j = 0; j < 3; ++j)  
    {  
        printf("%.2f  ", m[i][j]);  
    }  
    printf("\n");  
}  
}
```

```
float **add(float **m1, float **m2)
```

```
{  
    float **s = createNullMatrix();  
    for (int i = 0; i < 3; ++i)  
    {  
        for (int j = 0; j < 3; ++j)  
        {  
            s[i][j] = m1[i][j] + m2[i][j];  
        }  
    }  
    return s;  
}
```

```
float **subtract(float **m1, float **m2)
```

```
{  
    float **d = createNullMatrix();  
    for (int i = 0; i < 3; ++i)  
    {  
        for (int j = 0; j < 3; ++j)  
        {  
            d[i][j] = m1[i][j] - m2[i][j];  
        }  
    }  
}
```

```
    }  
    return d;  
}
```

```
float **multiply(float **m1, float **m2)
```

```
{  
    float **p = createNullMatrix();  
  
    for (int i = 0; i < 3; ++i)  
    {  
        for (int j = 0; j < 3; ++j)  
        {  
            for (int k = 0; k < 3; ++k)  
            {  
                p[i][j] += m1[i][k] * m2[k][j];  
            }  
        }  
    }  
    return p;  
}
```

```
int main()
```

```
{  
    float **a = input_matrix(), **b = input_matrix(), **p, **s, **d;  
  
    printf("\nMatrix A:\n");  
    print_matrix(a);  
  
    printf("\nMatrix B:\n");  
    print_matrix(b);
```

```

// adding matrices A and B: S = A + B

printf("\nMatrix S = A + B:\n\n");

s = add(a, b);

print_matrix(s);


// subtracting matrices A and B: D = A - B

printf("\nMatrix D = A - B:\n\n");

d = subtract(a, b);

print_matrix(d);


// multiplying matrices A and B: P = A x B

printf("\nMatrix P = A x B:\n\n");

p = multiply(a, b);

print_matrix(p);

return 0;
}

```

### OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q3.exe
Enter matrix :
1 2 3
4 5 6
7 8 9

Enter matrix :
7 8 9
4 5 6
1 2 3

Matrix A:
1.00  2.00  3.00
4.00  5.00  6.00
7.00  8.00  9.00

Matrix B:
7.00  8.00  9.00
4.00  5.00  6.00
1.00  2.00  3.00

Matrix S = A + B:
8.00  10.00  12.00
8.00  10.00  12.00
8.00  10.00  12.00

Matrix D = A - B:
-6.00  -6.00  -6.00
0.00  0.00  0.00
6.00  6.00  6.00

Matrix P = A x B:
18.00  24.00  30.00
54.00  69.00  84.00
90.00  114.00  138.00

```

**4. Write a program to find the range of a set of integers entered by the user. Range is the difference between the smallest and biggest number in the list.**

**Ans:**    CODE—

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

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("Enter size of array: ");
```

```
    scanf("%d", &n);
```

```
    int arr[n];
```

```
    printf("Enter %d array elements: ", n);
```

```
    for (int i = 0; i < n; ++i)
```

```
    {
```

```
        scanf("%d", &arr[i]);
```

```
    }
```

```
    int max = arr[0], min = arr[0];
```

```
    for (int i = 1; i < n; ++i)
```

```
    {
```

```
        if (arr[i] > max)
```

```
        {
```

```
            max = arr[i];
```

```
        }
```

```
        if (arr[i] < min)
```

```
        {
```

```
            min = arr[i];
```

```
        }
```

```
    }
```

```
printf("Range = Max - Min = %d - %d = %d\n", max, min, max - min);  
return 0;  
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q4.exe  
Enter size of array: 6  
Enter 6 array elements: 1 8 11 4 3 9  
Range = Max - Min = 11 - 1 = 10
```

**5. Write a C program which accepts ten integers from user and prints them in ascending order. Use array to store the integers.**

**Ans:** CODE—

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

```
void printArray(int a[], int n)
```

```
{  
    for (int i = 0; i < n; ++i)  
    {  
        printf("%d ", a[i]);  
    }  
}
```

```
void bubbleSort(int a[], int n)
```

```
{  
    for (int i = 0; i < n; ++i)  
    {  
        for (int j = 0; j < n - i - 1; ++j)  
        {  
            if (a[j] > a[j + 1])  
            {
```



```
        int tmp = a[j];
        a[j] = a[j + 1];
        a[j + 1] = tmp;
    }
}

printf("\nSorted array: ");
printArray(a, n);
}

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

    int arr[n];

    printf("Enter %d array elements: ", n);
    for (int i = 0; i < n; ++i)
    {
        scanf("%d", &arr[i]);
    }

    printf("Array entered: ");
    printArray(arr, n);

    bubbleSort(arr, n);
    return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q5.exe
Enter number of elements: 10
Enter 10 array elements: 1 9 8 10 19 3 0 2 7 12
Array entered: 1 9 8 10 19 3 0 2 7 12
Sorted array: 0 1 2 3 7 8 9 10 12 19
```

**6. Write a C program which accepts number obtained by five students in five subjects. Print the total marks obtained by all the students. Also determine the highest total marks.**

**Ans:** CODE –

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

```
int main()
```

```
{
```

```
    int rec[5][5];
```

```
    // accepting records of students
```

```
    for (int i = 0; i < 5; ++i)
```

```
    {
```

```
        printf("Enter marks of 5 subjects of students %d: ", i + 1);
```

```
        for (int j = 0; j < 5; ++j)
```

```
        {
```

```
            scanf("%d", &rec[i][j]);
```

```
        }
```

```
    }
```

```
    printf("\n");
```

```
    int max_total = 0, student;
```

```
    for (int i = 0; i < 5; ++i)
```

```
    {
```

```

int total = 0;
for (int j = 0; j < 5; ++j)
{
    total += rec[i][j];
}

printf("Total marks of student %d = %d\n", i + 1, total);

if (total > max_total)
{
    max_total = total;
    student = i + 1;
}
}

printf("\nHighest total marks obtained by student %d = %d\n", student, max_total);
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q6.exe
Enter marks of 5 subjects of students 1: 90 83 89 91 98
Enter marks of 5 subjects of students 2: 87 83 89 94 87
Enter marks of 5 subjects of students 3: 90 92 91 78 92
Enter marks of 5 subjects of students 4: 93 82 80 89 93
Enter marks of 5 subjects of students 5: 90 99 84 83 98

Total marks of student 1 = 451
Total marks of student 2 = 440
Total marks of student 3 = 443
Total marks of student 4 = 437
Total marks of student 5 = 454

Highest total marks obtained by student 5 = 454

```

**7. Write a C program which accepts roll numbers of ten students and marks obtained by them in five subjects and prints the names of the students who have obtained highest and second highest marks subject wise.**

**Ans:** CODE –

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

```
int main()
```

```
{
```

```
    int rec[5][5];
```

```
    // accepting records of students
```

```
    for (int i = 0; i < 5; ++i)
```

```
    {
```

```
        printf("Enter marks of 5 subjects of students %d: ", i + 1);
```

```
        for (int j = 0; j < 5; ++j)
```

```
        {
```

```
            scanf("%d", &rec[i][j]);
```

```
        }
```

```
    }
```

```
    printf("\n");
```

```
    int max_total = 0, second_max_total = 0, student, student1;
```

```
    // finding highest total marks
```

```
    for (int i = 0; i < 5; ++i)
```

```
    {
```

```
        int total = 0;
```

```
        for (int j = 0; j < 5; ++j)
```

```
        {
```

```
            total += rec[i][j];
```

```
        }
```

```
printf("Total marks of student %d = %d\n", i + 1, total);

if (total > max_total)
{
    max_total = total;
    student = i + 1;
}
}

// find second highest total marks
for (int i = 0; i < 5; ++i)
{
    int total = 0;
    for (int j = 0; j < 5; ++j)
    {
        total += rec[i][j];
    }

    if (total > second_max_total && total < max_total)
    {
        second_max_total = total;
        student1 = i + 1;
    }
}

printf("\nHighest total marks obtained by student %d = %d\n", student, max_total);
printf("Second highest total marks obtained by student %d = %d\n", student1, second_max_total);
return 0;
}
```

#### OUTPUT –

```
Enter marks of 5 subjects of students 1: 90 83 89 91 98
Enter marks of 5 subjects of students 2: 87 83 89 94 87
Enter marks of 5 subjects of students 3: 90 92 91 78 92
Enter marks of 5 subjects of students 4: 93 82 80 89 93
Enter marks of 5 subjects of students 5: 90 99 84 83 98

Total marks of student 1 = 451
Total marks of student 2 = 440
Total marks of student 3 = 443
Total marks of student 4 = 437
Total marks of student 5 = 454

Highest total marks obtained by student 5 = 454
Second highest total marks obtained by student 1 = 451
```

**8. Write a C program which accepts a matrix and prints its transpose.**

**Ans:** CODE –

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

```
#include <stdlib.h>
```

```
float **input_matrix()
```

```
{
```

```
    printf("\nEnter matrix :\n");
```

```
    float **mat;
```

```
    mat = (float **)malloc(3 * sizeof(float *));
```

```
    for (int i = 0; i < 3; ++i)
```

```
    {
```

```
        *(mat + i) = (float *)malloc(3 * sizeof(float));
```

```
        for (int j = 0; j < 3; ++j)
```

```
        {
```

```
            scanf("%f", &mat[i][j]);
```

```
        }
```

```
    }
```

```
    return mat;
}
```

```
void print_matrix(float **m)
{
    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            printf("%.2f  ", m[i][j]);
        }
        printf("\n");
    }
}
```

```
float **transpose(float **m)
{
    float **t;

    t = (float **)malloc(3 * sizeof(float *));
    for (int i = 0; i < 3; ++i)
    {
        *(t + i) = (float *)malloc(3 * sizeof(float));
    }

    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            t[j][i] = m[i][j];
        }
    }
}
```

```

    }
    return t;
}

int main()
{
    float **a;
    a = input_matrix();

    printf("\nMatrix A:\n");
    print_matrix(a);

    float **t = transpose(a);
    printf("\nTranspose of A:\n");
    print_matrix(t);

    return 0;
}

```

#### OUTPUT –

```

Enter matrix :
1 2 3
4 5 6
7 8 9

Matrix A:
1.00    2.00    3.00
4.00    5.00    6.00
7.00    8.00    9.00

Transpose of A:
1.00    4.00    7.00
2.00    5.00    8.00
3.00    6.00    9.00

```



**9. Write a C program to replace a square matrix by its transpose without using a second matrix.**

**Ans:** CODE –

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

```
int main(void)
```

```
{
```

```
    int row, col, iter;
```

```
    printf("Enter row and col of matrix: ");
```

```
    scanf("%d %d", &row, &col);
```

```
    // to allocate minimum number of cells(rows / columns) required for transpose
```

```
    if (row > col)
```

```
    {
```

```
        iter = row;
```

```
    }
```

```
    else
```

```
    {
```

```
        iter = col;
```

```
    }
```

```
    // initializing minimum-sized null matrix that is required to hold both original matrix and its transpose
```

```
    int m[iter][iter];
```

```
    for (int i = 0; i < iter; ++i)
```

```
    {
```

```
        for (int j = 0; j < iter; ++j)
```

```
        {
```

```
            m[i][j] = 0;
```

```
        }
```

```
    }
```

```
// getting matrix
printf("Enter matrix elements:\n");
for (int i = 0; i < row; ++i)
{
    for (int j = 0; j < col; ++j)
    {
        scanf("%d", &m[i][j]);
    }
}
```

```
// printing matrix
printf("\nMatrix entered:\n");
for (int i = 0; i < row; ++i)
{
    for (int j = 0; j < col; ++j)
    {
        printf("%d ", m[i][j]);
    }
    printf("\n");
}
```

```
// doing transpose
for (int i = 0; i < iter; ++i)
{
    for (int j = 0; j < i; ++j)
    {
        // swapping values
        int tmp = m[i][j];
        m[i][j] = m[j][i];
        m[j][i] = tmp;
    }
}
```

```

    }

    printf("\n");

    // printing final transpose
    printf("Transpose:\n");
    for (int i = 0; i < col; ++i)
    {
        for (int j = 0; j < row; ++j)
        {
            printf("%d ", m[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

#### OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q9.exe
Enter row and col of matrix: 3 4
Enter matrix elements:
1 2 3 4
2 3 4 5
3 4 5 6

Matrix entered:
1 2 3 4
2 3 4 5
3 4 5 6

Transpose:
1 2 3
2 3 4
3 4 5
4 5 6

```

**10. Consider the following procedure:**

- i. Take as input any four-digit number, using at least two different digits. (Leading zeros are allowed.)**
- ii. Arrange the digits in descending and then in ascending order to get two four-digit numbers, adding leading zeros if necessary.**
- iii. Subtract the smaller number from the bigger number. Let the difference be the new four-digit number.**
- iv. Go back to step ii.**

The above process, known as Kaprekar's routine, will always reach a fixed point (Known as Kaprekar Constant). Write a C-Code to implement the algorithm given above and find out the constant number. Also create an output file 'output.dat' in the working folder and write the following with appropriate format for each step of iteration: The 4-digit number, the larger number, the smaller number and the difference of the larger and the smaller number. Note: A.

The fixed point is achieved when in two consecutive steps the same number is obtained B.

In C the binary arithmetic operation  $m \% n$  gives the remainder when  $m$  is divided by  $n$ .

**Ans:** CODE—

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

```
void writeToFile(FILE *file, int original, int larger, int smaller, int difference)
{
    fprintf(file, "%d %d %d %d\n", original, larger, smaller, difference);
}
```

```
int kaprekarRoutine(int number)
{
    FILE *outputFile = fopen("output.dat", "w");

    if (outputFile == NULL)
    {
        printf("Error opening output file.\n");
        return -1;
    }
}
```

```
int count = 0;
int constant = 6174; // Kaprekar Constant
int currentNumber = number;
```

```
do
{
    int digits[4];
    int larger, smaller, difference;
```

```
    // Extract digits
```

```

    for (int i = 3; i >= 0; i--)
    {
        digits[i] = currentNumber % 10;
        currentNumber /= 10;
    }

    // Arrange digits in descending order
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4 - i - 1; j++)
        {
            if (digits[j] < digits[j + 1])
            {
                int temp = digits[j];
                digits[j] = digits[j + 1];
                digits[j + 1] = temp;
            }
        }
    }

    // Form larger and smaller numbers
    larger = digits[0] * 1000 + digits[1] * 100 + digits[2] * 10 + digits[3];
    smaller = digits[3] * 1000 + digits[2] * 100 + digits[1] * 10 + digits[0];

    // Calculate the difference
    difference = larger - smaller;

    // Write to output file
    writeToFile(outputFile, currentNumber, larger, smaller, difference);

    // Update current number for the next iteration
    currentNumber = difference;
    printf("number after iteration %d: %d\n", count + 1, currentNumber);
    count++;
} while (currentNumber != constant);

fclose(outputFile);

return count;
}

int main()
{
    int startingNumber;

    printf("Enter a four-digit number with at least two different digits: ");
    scanf("%d", &startingNumber);

    int steps = kaprekarRoutine(startingNumber);

    if (steps != -1)

```

```

{
    printf("Kaprekar Constant is reached in %d steps.\n", steps);
}

return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q10.exe
Enter a four-digit number with at least two different digits: 8713
number after iteration 1: 7353
number after iteration 2: 4176
number after iteration 3: 6174
Kaprekar Constant is reached in 3 steps.

```

#### output.dat File

0x output.dat X				
0x output.dat				
1	0	8731	1378	7353
2	0	7533	3357	4176
3	0	7641	1467	6174
4				

**11. Write a program which takes some numbers and computes the standard deviation of them.**

**Ans:** CODE –

```

#include <stdio.h>
#include <math.h>

float mean(int a[], int n)
{
    float s = 0.0;
    for (int i = 0; i < n; ++i)

```

```

    {
        s += a[i];
    }

    return s / n;
}

float stdDev(int a[], int n, float mean)
{
    float s1 = 0.0;
    for (int i = 0; i < n; ++i)
    {
        s1 += pow(a[i] - mean, 2);
    }

    return sqrt(s1 / n);
}

void print_array(int a[], int n)
{
    for (int i = 0; i < n; ++i)
    {
        printf("%d ", a[i]);
    }
    printf("\n");
}

int main()
{
    int size;
    printf("Enter size of array: ");

```

```
scanf("%d", &size);

int arr[size];

printf("Enter array elements: ");
for (int i = 0; i < size; ++i)
{
    scanf("%d", &arr[i]);
}

printf("Array entered: ");
print_array(arr, size);

printf("Standard deviation: %.3f\n", stdDev(arr, size, mean(arr, size)));
return 0;
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt4>q11.exe
Enter size of array: 8
Enter array elements: 3 8 1 10 9 8 7 11
Array entered: 3 8 1 10 9 8 7 11
Standard deviation: 3.219
```



## Assignment 5

1. Write a function to compute the distance between two points and use it to develop another function that will compute the area of the triangle whose vertices are A(x1, y1), B(x2, y2), and C(x3, y3). Use these functions to develop a function which returns a value 1 if the point (x, y) lies inside the triangle ABC, otherwise a value 0.

Ans: CODE –

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

```
#include <math.h>
```

```
typedef struct Point {
```

```
    float x;
```

```
    float y;
```

```
}Pt;
```

```
double distance(Pt *p1, Pt *p2) {
```

```
    return sqrt(pow(p1->x - p2->x, 2) + (p1->y - p2->y, 2));
```

```
}
```

```
double area(Pt *p1, Pt *p2, Pt *p3) {
```

```
    int x1 = p1->x, x2 = p2->x, x3 = p3->x, y1 = p1->y, y2 = p2->y, y3 = p3->y;
```

```
    return fabs((x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) * 0.5);
```

```
}
```

```
int checkPointInside(Pt *p1, Pt *p2, Pt *p3, Pt *p) {
```

```
    double a = area(p1, p2, p3);
```

```
    double a1 = area(p1, p2, p);
```

```
    double a2 = area(p1, p, p3);
```

```

        double a3 = area(p, p2, p3);

        // printf("%lf %lf %lf %lf\n", a, a1, a2, a3);

        if (a1 + a2 + a3 == a) {
            return 1;
        }
        else {
            return 0;
        }
    }

int main(void) {
    Pt p1, p2, p3, p;
    Pt *pt1 = &p1, *pt2 = &p2, *pt3 = &p3, *pt = &p;

    printf("Enter vertex 1 of triangle: ");
    scanf("%f %f", &(pt1->x), &(pt1->y));

    printf("Enter vertex 2 of triangle: ");
    scanf("%f %f", &(pt2->x), &(pt2->y));

    printf("Enter vertex 3 of triangle: ");
    scanf("%f %f", &(pt3->x), &(pt3->y));

    printf("Enter point to check: ");
    scanf("%f %f", &(pt->x), &(pt->y));

    int res = checkPointInside(pt1, pt2, pt3, pt);

```

```

    if (res == 1) {
        printf("res = %d. Point lies inside\n", res);
    }
    else {
        printf("res = %d. Point lies outside\n", res);
    }
    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q1.exe
Enter vertex 1 of triangle: 0 0
Enter vertex 2 of triangle: 5 0
Enter vertex 3 of triangle: 0 5
Enter point to check: 2 3
res = 1. Point lies inside

```

**2. A positive integer is entered through the keyboard, write a C function to find the binary equivalent of this number using recursion.**

**Ans:** CODE –

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

```
int s = 0, p = 1;
```

```

int binary(int n) {
    if (n == 0) {
        return s;
    }
    else {
        s += (n % 2) * p;
        p *= 10;
        return binary(n / 2);
    }
}

```

```

}

int main() {
    int n, bin;

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

    bin = binary(n);

    printf("Binary: %d\n", bin);

    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q2.exe
Enter decimal: 100
Binary: 1100100

```

**3. Write a recursive function which returns the sum of individual digits of a number passed as argument.**

**Ans:** CODE—

```

#include <stdio.h>

int s = 0;

int sumDigits(int n) {
    if (!n) {
        return s;
    }
    else {
        s += n % 10;
        return sumDigits(n / 10);
    }
}

```

```

int main(void) {
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    printf("Sum of digits of %d is %d\n", n, sumDigits(n));
    return 0;
}

```

OUTPUT—

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q3.exe
Enter number: 2142
Sum of digits of 2142 is 9

```

**4. Amicable numbers are a pair of numbers with the following property: the sum of all of the proper divisors of the first number (not including itself) exactly equals the second number while the sum of all of the proper divisors of the second number (not including itself) likewise equals the first number. Write a function which takes as argument two numbers and returns 1 if they are amicable numbers else 0.**

**Ans:** CODE —

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

```

int checkAmicable(int x, int y)
{
    int sx = 0, sy = 0;

    for (int i = 1; i <= x / 2; ++i)
    {
        if (x % i == 0)
        {
            sx += i;
        }
    }
}

```

```
}
```

```
for (int j = 1; j <= y / 2; ++j)
```

```
{
```

```
    if (y % j == 0)
```

```
    {
```

```
        sy += j;
```

```
    }
```

```
}
```

```
if (sx == y && sy == x)
```

```
{
```

```
    return 1;
```

```
}
```

```
else
```

```
{
```

```
    return 0;
```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
    int n1, n2;
```

```
    printf("Enter n1 and n2: ");
```

```
    scanf("%d %d", &n1, &n2);
```

```
    int res = checkAmicable(n1, n2);
```

```
    if (res)
```

```
    {
```

```
        printf("%d\n", res);
```

```
}  
else  
{  
    printf("%d\n", res);  
}  
return 0;  
}
```

OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q4.exe  
Enter n1 and n2: 220 284  
1
```

**5. Write a C program which accepts a string from user and counts the number of characters in the string without using string library functions.**

**Ans:** CODE –

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

```
int countCharacters(const char s[])
```

```
{  
    int c = 0;  
    for (int i = 0; s[i] != '\0'; ++i)  
    {  
        c += 1;  
    }  
    return c;  
}
```

```

int main(void)
{

    char string[100];

    printf("Enter string: ");

    scanf("%s", &string);

    printf("Number of characters in string is %d\n", countCharacters(string));

    return 0;
}

```

OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q5.exe
Enter string: basketball
Number of characters in string is 10

```

**6. Write a C program which accepts a string from user and prints the reverse of the string without using string library functions.**

**Ans:** CODE –

```

#include <stdio.h>

int length(const char s[])
{
    int l = 0;
    for (int i = 0; s[i] != '\0'; ++i)
    {
        l += 1;
    }
    return l;
}

```



```
int main()
{
    char s[100], revs[100];
    int i;

    printf("Enter string: ");
    scanf("%[^\\n]c", s);

    int len = length(s);

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

    for (i = 0; i < len; ++i)
    {
        revs[i] = s[len - i - 1];
    }
    revs[i] = '\\0';
    printf("Reversed String: %s\\n", revs);
    return 0;
}
```

#### OUTPUT –

```
D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q6.exe
Enter string: basketball
Original string: basketball
Reversed String: llabteksab
```

**7. Write a C program which accepts a full name from user prints the initials. E.g. SRT for Sachin Ramesh Tendulkar.**

**Ans:** CODE –

```
#include <stdio.h>

int main()
{
    char s[100], initials[100];
    int j = 1;

    printf("Enter string: ");
    scanf("%[^\\n]c", s);

    initials[0] = s[0];
    for (int i = 1; s[i] != '\\0'; ++i)
    {
        if (s[i] == ' ')
        {
            initials[j++] = s[i + 1];
        }
    }
    initials[j] = '\\0';
    printf("%s\\n", initials);
    return 0;
}
```

OUTPUT –

```
D:\\personal\\btech-JU\\assignments\\sem1\\CPNM Lab\\asgmt5>q7.exe
Enter string: Sachin Ramesh Tendulkar
SRT
```

**8. Write a C program which accepts any string of the form:**

**“Ustad Bade Ghulam Ali Khan was the Tansen of the 20th century”**

**and prints it as**

**“Ustad|Bade|Ghulam|Ali|Khan|was|the|Tansen|of|the|20th|century”.**

**Ans: CODE—**

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

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    char ch;
```

```
    char *s1 = (char *)malloc(100 * sizeof(char));
```

```
    int len = 0;
```

```
    printf("Enter string: ");
```

```
    while (1)
```

```
    {
```

```
        scanf("%c", &ch);
```

```
        if (ch == '\n')
```

```
        {
```

```
            break;
```

```
        }
```

```
        else
```

```
        {
```

```
            s1[len++] = ch;
```

```
        }
```

```
    }
```

```
    s1[len] = '\0';
```

```

char *s2 = (char *)malloc(len * sizeof(char));

int i;

for (i = 0; s1[i] != '\0'; ++i)
{
    if (s1[i] == ' ')
    {
        s2[i] = '|';
    }
    else
    {
        s2[i] = s1[i];
    }
}

s2[i] = '\0';

printf("Modified string: %s\n", s2);

free(s1);

free(s2);

s1 = NULL;

s2 = NULL;

return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q8.exe
Enter string: Ustad Bade Ghulam Ali Khan was the Tansen of the 20th century
Modified string: Ustad|Bade|Ghulam|Ali|Khan|was|the|Tansen|of|the|20th|century

```

**9. Write a program that reads a line and converts it into all capitals without using any string library function.**

**Ans:** CODE –

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

```
#include <stdlib.h>
```

```
char *inputString()
```

```
{
```

```
    char *s = (char *)malloc(100 * sizeof(char));
```

```
    char ch;
```

```
    int i = 0;
```

```
    printf("Enter string: ");
```

```
    while (1)
```

```
    {
```

```
        scanf("%c", &ch);
```

```
        if (ch == '\n')
```

```
        {
```

```
            break;
```

```
        }
```

```
        else
```

```
        {
```

```
            s[i++] = ch;
```

```
        }
```

```
    }
```

```
    s[i] = '\0';
```

```
    s = (char *)realloc(s, i * sizeof(char));
```

```
        return s;
    }

int getLength(char *s)
{
    int len = 0;
    while (s[len] != '\0')
    {
        len++;
    }
    return len;
}

int main(void)
{
    char *s = inputString();
    int len = getLength(s);

    char *ns = (char *)malloc(len * sizeof(char));

    for (i = 0; s[i] != '\0'; ++i)
    {
        if ((int)s[i] >= 97 && (int)s[i] <= 122)
        {
            ns[i] = (char)((int)s[i] - 32);
        }
        else
        {
            ns[i] = s[i];
        }
    }
}
```

```

        ns[i] = '\0';

        printf("String in upper case: %s\n", ns);

        free(s);

        free(ns);

        s = NULL, ns = NULL;

        return 0;

    }

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q9.exe
Enter string: Ustad Bade Ghulam Ali Khan was the Tansen of the 20th century
String in upper case: USTAD BADE GHULAM ALI KHAN WAS THE TANSEN OF THE 20TH CENTURY

```

**10. Write a program to count the number of occurrences of any two vowels in succession in a line of text.**

**Ans:** CODE –

```

#include <stdio.h>

#include <stdlib.h>

char *inputString()
{
    char *s = (char *)malloc(50 * sizeof(char));

    char ch;

    int i = 0;

    printf("Enter string: ");

    while (1)
    {
        scanf("%c", &ch);
    }
}

```

```
        if (ch == '\n')
        {
            break;
        }
        else
        {
            s[i++] = ch;
        }
    }
    s[i] = '\0';

    s = (char *)realloc(s, i * sizeof(char));
    return s;
}
```

```
int isVowel(char c)
```

```
{
    if (c == 'A' || c == 'a' || c == 'E' || c == 'e' || c == 'O' || c == 'o' || c == 'I' || c == 'i' || c == 'u' || c == 'U')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
int countConsecutiveVowels(char *s)
```

```
{
```



```

    int i = 1, c = 0;
    while (s[i] != '\0')
    {
        if (isVowel(s[i - 1]) && isVowel(s[i]))
        {
            c += 1;
        }
        i++;
    }
    return c;
}

int main(void)
{
    char *s = inputString();
    printf("There are %d consecutive vowels in the string.", countConsecutiveVowels(s));
    free(s);
    s = NULL;
    return 0;
}

```

#### OUTPUT –

```

D:\personal\btech-JU\assignments\sem1\CPNM Lab\asgmt5>q10.exe
Enter string: aakaekjasdoeusdak
There are 4 consecutive vowels in the string.

```