Assignment-3

Code:

```c
// Q1.  Write a C program for printing prime factors of a number using functions.
#include <stdio.h>
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i==0)
            return 0;
        else if (i+1==x)
            return x;
    }
}
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    printf("Prime Factors: ");
    for (int i=2;i<=n;i++)
    {
        if (n%i==0 && prime_number(i))
            printf("%d ",i);
    }
    return 0;
}
```

Output:

```
Enter Number: 13
Prime Factors: 13
```

Code:

```c
// Q2. Write a C program for printing prime numbers between a interval.
#include <stdio.h>
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i == 0)
            return 0;
        else if (i+1 == x)
            return x;
    }
}
int main()
{
    int a,b;
    printf("Enter Starting Number: ");
    scanf("%d",&a);
    printf("Enter Ending Number: ");
    scanf("%d",&b);
    printf("Prime Numbers between %d and %d: ",a,b);
    if (a<2)
        a=2;
    for (int i=a;i<=b;i++)
    {
    int c=prime_number(i);
    if (c!=0)
        printf("%d ",i);
    }
    return 0;
}
```

Output:

```
Enter Starting Number: 8
Enter Ending Number: 26
Prime Numbers between 8 and 26: 11 13 17 19 23
```

Code:

```c
// Q3. Write a C program to convert decimal to binary numbers & vice versa.
#include <stdio.h>
#include <math.h>
int Decimal_to_Binary(int x)
{
    int A[64],i=0;
    printf("Decimal: %d --> Binary: ",x);
    if (x == 0)
        printf("0");
    while (x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
int Binary_to_Decimal(int x)
{
    int A[128],i=0,s=0;
    printf("Binary: %d --> Decimal: ",x);
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i] == 1)
            s+=A[i]*pow(2,i);
        i++;

    }
    printf("%d",s);
}
```

```c
int main()
{
    int n,i;
    printf("Enter\n1. for Decimal To Binary, or\n2. for Binary to Decimal: ");
    scanf("%d",&n);
    switch(n)
    {
        case 1:
        {
            printf("Enter Decimal Number: ");
            scanf("%d",&i);
            Decimal_to_Binary(i);
            break;
        }
        case 2:
        {
            printf("Enter Binary Number (Only 0 and 1): ");
            scanf("%d",&i);
            Binary_to_Decimal(i);
            break;
        }
        default:
        {
            printf("ERROR!! Enter from above choices.");
        }
    }
    return 0;
}
```

Output:

```
Enter
1. for Decimal To Binary, or
2. for Binary to Decimal: 1
Enter Decimal Number: 13
Decimal: 13 --> Binary: 1101
```

```
Enter
1. for Decimal To Binary, or
2. for Binary to Decimal: 2
Enter Binary Number (Only 0 and 1): 111001
Binary: 111001 --> Decimal: 57
```

Code:

```c
// Q4. Write a C program to convert octal to binary & vice versa.
#include <stdio.h>
#include <math.h>
void Decimal_to_Binary(int x)
{
    int A[64],i=0;
    while(x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
int Binary_to_Decimal(int x)
{
    int A[64],i=0,s=0;
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i]==1)
            s+=A[i]*pow(2,i);
        i++;
    }
    return s;
}
```

```c
int main()
{
    int c,n;
    printf("Enter\n1.for Octal to Binary, or\n2. for Binary to Octal: ");
    scanf("%d",&c);
    switch(c)
    {
        case 1:
        {
            char A[16];
            int d;
            printf("Enter Octal Number: ");
            scanf("%s",A);
            sscanf(A,"%o",&n);
            printf("Octal: %s --> Binary: ",A);
            Decimal_to_Binary(n);
            break;
        }
        case 2:
        {
            char A[16];
            printf("Enter Binary Number (only 0 and 1): ");
            scanf("%d",&n);
            int d=Binary_to_Decimal(n);
            sprintf(A,"%o",d);
            printf("Binary: %d --> Octal: %s",n,A);
            break;
        }
    }
}
```

Output:

```
Enter
1.for Octal to Binary, or
2. for Binary to Octal: 1
Enter Octal Number: 17
Octal: 17 --> Binary: 1111
```

```
Enter
1.for Octal to Binary, or
2. for Binary to Octal: 2
Enter Binary Number (only 0 and 1): 110011
Binary: 110011 --> Octal: 63
```

Code:

```c
// Q5. Write a C program to find mean, median, mode.
#include <stdio.h>
float mean(float A[], int x)
{
    float s = 0;
    for (int i = 0; i < x; i++)
        s += A[i];
    return s / x;
}
float median(float A[], int x)
{
    if (x % 2 == 0)
        return (A[x / 2] + A[x / 2 - 1]) / 2;
    else
        return A[x / 2];
}
void sort_array_increasing(float A[], int x)
{
    float t;
    for (int i = 0; i < x; i++)
    {
        for (int j = i + 1; j < x; j++)
        {
            if (A[i] > A[j])
            {
                t = A[i];
                A[i] = A[j];
                A[j] = t;
            }
        }
    }
}
```

```c
float mode(float A[], int x)
{
    int max_count = 0, count;
    float mode_value = A[0];
    for (int i = 0; i < x; i++)
    {
        count = 1; // Count occurrences of A[i]
        for (int j = i + 1; j < x; j++)
        {
            if (A[i] == A[j])
                count++;
        }

        if (count > max_count)
        {
            max_count = count;
            mode_value = A[i];
        }
    }
    if (max_count == 1)
    {
        printf("No mode exists (all values are unique).\n");
        return -1;
    }
    return mode_value;
}
```

```c
int main()
{
    int n;
    printf("Enter Number of Elements: ");
    scanf("%d", &n);
    float Element[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter Element %d: ", i + 1);
        scanf("%f", &Element[i]);
    }
    sort_array_increasing(Element, n);
    printf("Elements in increasing order are: ");
    for (int i = 0; i < n; i++)
        printf("%.2f ", Element[i]);
    printf("\n");
    printf("Mean: %.2f\n", mean(Element, n));
    printf("Median: %.2f\n", median(Element, n));
    float mode_result = mode(Element, n);
    if (mode_result != -1)
        printf("Mode: %.2f\n", mode_result);
    return 0;
}
```

Output:

```
Enter Number of Elements: 5
Enter Element 1: 3
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 5
Enter Element 5: 1
Elements in increasing order are: 1.00 2.00 3.00 3.00 5.00
Mean: 2.80
Median: 3.00
Mode: 3.00
```

Assignment-4

Code:

```c
// Q6. Write a program in C to make such a pattern like a right angle triangle with a number, which will repeat a number in a row.
#include <stdio.h>
int main()
{
    int r;
    printf("Enter Rows: ");
    scanf("%d",&r);
    for (int i=1;i<=r;i++)
    {
        for (int j=1;j<=i;j++)
            printf("%d",i);
        printf("\n");
    }
}
```

Output:

```
Enter Rows: 5
1
22
333
4444
55555
```

Code:

```c
// Q7 Write a program in C to make a pyramid pattern with
numbers increased by 1.
#include <stdio.h>
int main()
{
    int r;
    printf("Enter Rows: ");
    scanf("%d",&r);
    int a=r*(r+1)/2;
    int A[a];
    for (int i=0;i<a;i++)
        A[i]=i+1;
    int t=0;
    for (int i=0;i<r;i++)
    {
        for (int j=0;j<=i;j++)
        {
            printf("%d ",A[t]);
            t++;
        }
        printf("\n");
    }
}
```

Output:

```
Enter Rows: 4
1
2 3
4 5 6
7 8 9 10
```

Code:

```c
// Q8. Write a program in C to print Floyd's Triangle.
#include <stdio.h>
int main()
{
    int r;
    printf("Enter number of rows: ");
    scanf("%d",&r);
    for (int i=1;i<=r;i++)
    {
        int s=i%2;
        for (int j=1;j<=i;j++)
        {
            printf("%d ",s);
            s=1-s;
        }
        printf("\n");
    }
}
```

Output:

```
Enter number of rows: 6
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

Code:

```c
// Q9. Write a program in C to display a pattern like a
diamond.
#include <stdio.h>
int main()
{
    int rows;

    printf("Enter the number of rows (odd number): ");
    scanf("%d", &rows);

    int mid = (rows + 1) / 2;
    for (int i = 1; i <= mid; i++)
    {
        for (int j = 1; j <= mid - i; j++)
            printf(" ");
        for (int j = 1; j <= 2 * i - 1; j++)
            printf("*");
        printf("\n");
    }
    for (int i = mid - 1; i >= 1; i--)
    {
        for (int j = 1; j <= mid - i; j++)
            printf(" ");
        for (int j = 1; j <= 2 * i - 1; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

Output:

```
Enter the number of rows (odd number): 7
   *
  ***
 *****
*******
 *****
  ***
   *
```

Code:

```c
// Q10.  Write a program in C to display a pattern like a
diamond.
#include <stdio.h>
void printPascalsTriangle(int rows)
{
    int arr[rows][rows];
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j <= i; j++)
        {
            if (j == 0 || j == i)
                arr[i][j] = 1;
            else
                arr[i][j] = arr[i - 1][j - 1] + arr[i - 1]
                [j];
        }
    }
    for (int i = 0; i < rows; i++)
    {
        for (int space = 0; space < rows - i - 1; space++)
            printf(" ");
        for (int j = 0; j <= i; j++)
            printf("%d ", arr[i][j]);
        printf("\n");
    }
}
```

```c
int main() {
    int n;
    printf("Enter the number of rows for Pascal's Triangle:
    ");
    scanf("%d", &n);
    if (n <= 0)
        printf("Number of rows must be a positive integer.
        \n");
    else
        printPascalsTriangle(n);
    return 0;
}
```

Output:

```
Enter the number of rows for Pascal's Triangle: 5
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

Assignment-5

Code:

```c
// Q11. Write a program in C to check whether a number can be expressed as the sum of two prime numbers
#include <stdio.h>
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i == 0)
            return 0;
        else if (i+1 == x)
            return x;
    }
}
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    int a = n / 2;
    for (int i = 2; i <= a; i++)
    {
        if (prime_number(i) && prime_number(n - i))
            printf("%d = %d + %d\n", n, i, n - i);
    }
    return 0;
}
```

Output:

```
Enter Number: 26
26 = 3 + 23
26 = 7 + 19
26 = 13 + 13
```

Code:

```c
// Q12. Write a C program to check whether a number is a
Armstrong Number or not.
#include <stdio.h>
#include <math.h>
int length(int x)
{
    int len =0;
    while (x>0)
    {
        x=x/10;
        len++;
    }
    return len;
}
void Armstrong_Number(int x)
{
    int l=length(x);
    int A[l],s=0,a=x;
    while (a>0)
    {
        int r=a%10;
        s=s+round(pow(r,l));
        a/=10;
    }
    if (s==x)
        printf("Number %d is an Armstrong Number.\n",x);
    else
        printf("Number %d is NOT an Armstrong Number.\n",x);
}
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    Armstrong_Number(n);
    return 0;
}
```

Output:

```
Enter Number: 8208
Number 8208 is an Armstrong Number.
```

```
Enter Number: 562
Number 562 is NOT an Armstrong Number.
```

Code:

```c
// Q13. Write C program to convert binary to hexadecimal &
vice versa
#include <stdio.h>
#include <math.h>
int Binary_to_Decimal(int x)
{
    int A[32],i=0,s=0;
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i]==1)
            s+=A[i]*pow(2,i);
        i++;

    }
    return s;
}
int Decimal_to_Binary(int x)
{
    int A[64],i=0;
    if (x==0)
        return 0;
    while (x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
```

```c
int main()
{
    int c,n;
    printf("Enter\n1. for Binary to Hexadecimal, or\n2.
    Hexadecimal to Binary: ");
    scanf("%d",&c);
    switch(c)
    {
        case 1:
        {
            printf("Enter Binary Number: ");
            scanf("%d",&n);
            int d=Binary_to_Decimal(n);
            printf("Binary: %d --> Hexadecimal: %X\n",d,d);
            break;
        }
        case 2:
        {
            char A[16],B[32];
            printf("Enter Hexadecimal Number: ");
            scanf("%s",&A);
            sscanf(A,"%X",&n);
            printf("Hexadecimal: %s --> Binary: ",A);
            Decimal_to_Binary(n);
            break;
        }
    }
}
```

Output:

```
Enter
1. for Binary to Hexadecimal, or
2. Hexadecimal to Binary: 1
Enter Binary Number: 11011
Binary: 27 --> Hexadecimal: 1B
```

```
Enter
1. for Binary to Hexadecimal, or
2. Hexadecimal to Binary: 2
Enter Hexadecimal Number: 11E
Hexadecimal: 11E --> Binary: 100011110
```

Code:

```c
// Q14. Write a program in C to find the number and sum of all integers between 100 and 200 which are divisible by 9.
#include <stdio.h>
int main()
{
    int s=0;
    printf("Number divisible by 9 are: \n");
    for (int i=100;i<=200;i++)
    {
        if (i%9 == 0)
        {
            printf("%d ",i);
            s+=i;
        }
    }
    printf("\nSum: %d",s);
}
```

Output:

```
Number divisible by 9 are:
108 117 126 135 144 153 162 171 180 189 198
Sum: 1683
```

Code:

```c
// Q15. Write a program to find out the second largest factor of a
number. Eg. Input->100, Output-> 50.
#include <stdio.h>
int main()
{
    int n,index=0;
    printf("Enter Number: ");
    scanf("%d",&n);
    int A[n/2];
    for (int i=1;i<=n;i++)
    {
        if (n%i == 0)
        {
            A[index]=i;
            index++;
        }
    }
    printf("Second largest Factor of %d is %d.",n,A[index-2]);
}
```

Output:

```
Enter Number: 75
Second largest Factor of 75 is 25.
```

Assignment-6

Code:

```c
// Q16. Write C Program to Find the Largest Element in an Array.
#include <stdio.h>
int main()
{
    int n;
    printf("Enter Size of an Array: ");
    scanf("%d",&n);
    int A[n];
    for (int i=0;i<n;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    int t;
    for (int i=0;i<n;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (A[i] < A[j])
            {
                t=A[i];
                A[i]=A[j];
                A[j]=t;
            }
        }
    }
    printf("The Largest Element in Array is %d.",A[0]);
}
```

Output:

```
Enter Size of an Array: 6
Enter Element 1: 3
Enter Element 2: 2
Enter Element 3: -1
Enter Element 4: 2
Enter Element 5: 9
Enter Element 6: 8
The Largest Element in Array is 9.
```

Code:

```c
// Q17. Write a C Program to Find the Maximum and Minimum in an Array.
#include <stdio.h>
int main()
{
    int n;
    printf("Enter Size of an Array: ");
    scanf("%d",&n);
    int A[n];
    for (int i=0;i<n;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    int t;
    for (int i=0;i<n;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (A[i] < A[j])
            {
                t=A[i];
                A[i]=A[j];
                A[j]=t;
            }
        }
    }
    printf("The Largest Element in an Array is %d.\n",A[0]);
    printf("The Smallest Element in an Array is %d.\n",A[n-1]);
}
```

Output:

```
Enter Size of an Array: 5
Enter Element 1: 2
Enter Element 2: -5
Enter Element 3: -6
Enter Element 4: 7
Enter Element 5: 10
The Largest Element in an Array is 10.
The Smallest Element in an Array is -6.
```

Code:

```c
// Q18. Write C Program to Search an Element in an Array (Binary search)
#include <stdio.h>
int main()
{
    int a,b,t;
    printf("Enter Size of Array: ");
    scanf("%d",&a);
    int A[a];
    for (int i=0;i<a;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    printf("Enter Number to search: ");
    scanf("%d",&b);
    int found=0;
    for (int i=0;i<a;i++)
    {
        if (b==A[i])
        {
            printf("Element %d Found at Index %d.\n",b,i);
            found=1;
        }
    }
    if (!found)
        printf("Element %d NOT Found.",b);
}
```

Output:

```
Enter Size of Array: 6
Enter Element 1: 1
Enter Element 2: 3
Enter Element 3: 6
Enter Element 4: -7
Enter Element 5: 13
Enter Element 6: 9
Enter Number to search: 6
Element 6 Found at Index 2.
```

Code:

```c
// Q19. Write a C Program to find all pairs in array of integers whose sum is equal to given number.
#include <stdio.h>
int main()
{
    int A[] = {0,1,2,3,4,5,6,7,8,9};
    int n,c=0;
    printf("Enter Element Number: ");
    scanf("%d",&n);
    for (int i=0;i<9;i++)
    {
        for (int j=i+1;j<9;j++)
        {
            if (n == A[i]+A[j])
            {
                printf("The pair whose sum is %d are %d and %d.\n",n,A
                [i],A[j]);
                c++;
            }
        }
    }
    if (c == 0)
        printf("No pairs found.");
    return 0;
}
```

Output:

```
Enter size of an array: 5
Enter Element 1: 6
Enter Element 2: 3
Enter Element 3: -4
Enter Element 4: 2
Enter Element 5: 7
Enter Element Number: 9
The pair whose sum is 9 are 6 and 3.
The pair whose sum is 9 are 2 and 7.
```

Code:

```c
// Q20. Write a C program for calculating the factorial of a number.
#include <stdio.h>
int main()
{
    int n,f=1;
    printf("Enter Number: ");
    scanf("%d",&n);
    for (int i=n;i>=2;i--)
        f=f*i;
    printf("Factorial of %d is %d.",n,f);
    return 0;
}
```

Output:

```
Enter Number: 6
Factorial of 6 is 720.
```

Assignment-7

Code:

```c
// Q21. Write a C program to generate fibonacci numbers and to find out if a
given number is a fibonacci number or not.

#include <stdio.h>
int main()
{
    int n,c=0,t;
    printf("Enter Number of terms of Fibonacci Series: ");
    scanf("%d",&t);
    int A[t];
    A[0]=0,A[1]=1;
    for (int i=2;i<t;i++)
        A[i] = A[i-1]+A[i-2];
    printf("Fibonacci Series: ");
    for (int i=0;i<t;i++)
        printf("%d ",A[i]);
    printf("\nEnter Number to Search in fibonacci Series: ");
    scanf("%d",&n);
    for (int i=0;i<t;i++)
    {
        if (n == A[i])
        {
            printf("%d is a Fibonacci Number.",n);
            c++;
            break;
        }
    }
    if (c == 0)
        printf("%d is not a Fibonacci Number.",n);
    return 0;
}
```

Output:

```
Enter Number of terms of Fibonacci Series: 14
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233
Enter Number to Search in fibonacci Series: 143
143 is not a Fibonacci Number.
```

Code:

```c
// Q22. Write C programs to check, if an integer is a power of 2 or not in a
single line.

#include <stdio.h>
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    printf("%d is %s a power if 2.\n",n,(n>0 && (n & (n-1)) == 0 ) ? "" :
    "not" );
    return 0;
}
```

Output:

```
Enter Number: 32
32 is  a power if 2.
```

```
Enter Number: 126
126 is not a power if 2.
```

Code:

```c
// Q23. Write code to remove duplicates in a sorted array.

#include <stdio.h>
int main()
{
    int A[] = {1,2,3,3,4,4,5,6,7,7}; //Sorted Array (increasing order)
    printf("Array before removing duplicates is: ");
    for (int i=0;i<10;i++)
        printf("%d ",A[i]);
    for (int i=0;i<10;i++)
    {
        for (int j=i+1;j<10;j++)
        {
            if (A[i] == A[j])
                A[j]=0;
        }
    }
    printf("\nArray after removing duplicates is: ");
    for (int i=0;i<10;i++)
    {
        if (A[i] != 0)
            printf("%d ",A[i]);
    }
}
```

Output:

```
Array before removing duplicates is: 1 2 3 3 4 4 5 6 7 7
Array after removing duplicates is: 1 2 3 4 5 6 7
```

Code:

```c
// Q25. Write C code to return a string from a function.
#include <stdio.h>
#include <string.h>
int function_string(char str[],int size)
{
    printf("Enter String: ");
    fgets(str,size,stdin);
}
int main()
{
    char str[100];
    function_string(str,sizeof(str));
    printf("Returned String is: %s",str);
}
```

Output:

```
Enter String: Let us C!!
Returned String is: Let us C!!
```

Assignment-8

Code:

```c
// Q27. Write a C program to multiply two matrices.

#include <stdio.h>
int main()
{
    int r1,r2,c1,c2;
    printf("Enter Rows and Columns of First Matrix: ");
    scanf("%d %d",&r1,&c1);
    int A[r1][c1];
    // Insert Elements of First Matrix.
    for (int i=0;i<r1;i++)
    {
        for (int j=0;j<c1;j++)
        {
            printf("Enter Element %d%d: ",i,j);
            scanf("%d",&A[i][j]);
        }
    }
    // Printing Element of First Matrix
    printf("Element of First Matrix %dx%d are: \n",r1,c1);
    for (int i=0;i<r1;i++)
    {
        for (int j=0;j<c1;j++)
            printf("%d ",A[i][j]);
        printf("\n");
    }
    printf("Enter Rows and Columns of Second Matrix: ");
    scanf("%d %d",&r2,&c2);
    int B[r2][c2];
```

Output:

```
Element of First Matrix 2x3 are:
2 4 -1
6 10 15
Enter Rows and Columns of Second Matrix: 3 1
Enter Element 00: 6
Enter Element 10: 2
Enter Element 20: 4
Element of Second Matrix 3x1 are:
6
2
4
Element of Resultant Matrix 2x1 are:
16
116
```

Code:

```c
// Q28. Write a C program to check for palindromes.
#include <stdio.h>
void check_palindrome(int number)
{
    int original_number = number;
    int reversed=0;
    while (number > 0)
    {
        int remainder = number%10;
        reversed = reversed * 10 + remainder;
        number/=10;
    }
    if (original_number == reversed)
        printf("Yes!! %d is a Palindrome.\n",original_number);
    else
        printf("No!! %d is NOT a Palindrome.\n",original_number);
}
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    check_palindrome(n);
    return 0;
}
```

Output:

```
Enter Number: 1234321
Yes!! 1234321 is a Palindrome.
```

Code:

```c
// Q29. Write a C program to convert a decimal number into a binary number.
#include <stdio.h>
int main()
{
    int n,length=0;
    printf("Enter Decimal Number: ");
    scanf("%d",&n);
    int original_number=n,binary=0,reversed_binary=0;
    while(n > 0)
    {
        int remainder=n%2;
        reversed_binary = reversed_binary * 10 + remainder;
        n/=2;
    }
    while (reversed_binary > 0)
    {
        int remainder = reversed_binary%10;
        binary = binary * 10 + remainder;
        reversed_binary/=10;
    }
    printf("Decimal: %d --> Binary: %d\n",original_number,binary);
}
```

Output:

```
Enter Decimal Number: 13
Decimal: 13 --> Binary: 1101
```

Code:

```c
// Q30. Write a program to check, if a given year is a leap year or not.
#include <stdio.h>
int main()
{
    int n;
    printf("Enter Year: ");
    scanf("%d",&n);
    if ((n%400 == 0) || (n%4 == 0 && n%100 != 0))
        printf("Yes!! %d is a Leap Year.\n",n);
    else
        printf("No!! %d is NOT a Leap Year.\n",n);
}
```

Output:

```
Enter Year: 2024
Yes!! 2024 is a Leap Year.
```

Assignment-9

Code:

```c
// Q31. Write your own sqrt function in C.
#include <stdio.h>
#include <math.h>
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    printf("The Root of %d is %.3f.\n",n,pow(n,0.5));
}
```

Output:

```
Enter Number: 81
The Root of 81 is 9.000.
```

Code:

```c
// Q33. How can we sum the digits of a given number in single statement.
#include <stdio.h>
int main()
{
    int a,s=0;
    printf("Enter Number: ");
    scanf("%d",&a);
    int n=a;
    for(; n>0; s+=n%10, n/=10);
    printf("The sum of digits of number %d is %d.\n",a,s);
    return 0;
}
```

Output:

```
Enter Number: 956
The sum of digits of number 956 is 20.
```

Code:

```c
// Q35. Write a C program generate permutations.
#include <stdio.h>
int factorial(int n)
{
    int f=1;
    for (int i=n;i>=2;i--)
        f=f*i;
}
int main()
{
    int n,r;
    printf("Enter the value of 'n' and 'r' nPr: ");
    scanf("%d %d",&n,&r);
    printf("nPr for %dP%d is %d.",n,r,factorial(n)/factorial(n-r));
}
```

Output:

```
Enter the value of 'n' and 'r' nPr: 6
3
nPr for 6P3 is 120.
```