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
```