Raunak Gayen
2020EEB066

1.a Compute X^n, where X is any valid number and n is an integer value.

```c
#include<stdio.h>

long int power(long int n, long int p)
{
    if(p==0)
        return 1;
    else if(p==1)
        return n;
    else
        return n*power(n,p-1);
}
int main()
{
    long int ans,num,p;
    printf(" Enter any number \n");
    scanf("%ld",&num);
    printf(" Enter power \n");
    scanf("%ld",&p);
    ans = power(num,p);
    printf(" %ld raised to the power %ld is ---  %ld\n",num,p,ans);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
 Enter any number
6
 Enter power
3
 6 raised to the power 3 is ---  216
```

1.b.i Swap values of two integer variables with (a) using a temporary variable,

```c
#include<stdio.h>
void swap_temp(int a,int b)
{
    int c;
    c=b;
    b=a;
    a=c;

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

int main()
{
    int a,b;
    printf(" Enter the values of a and b \n");
    scanf("%d%d",&a,&b);
    printf(" Before swapping a = %d and b = %d \n\n",a,b);
    swap_temp(a,b);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
 Enter the values of a and b
5 10
 Before swapping a = 5 and b = 10

 After swapping a = 10 and b = 5
```

1.b.ii Swap values of two integer variables without using a temporary variable.

```c
#include<stdio.h>
void swap_without_temp(int a,int b)
{

    a=a+b;  //suppose a=10 b=20 so a = a+b --> a becomes 30
    b=a-b;  // now b = 30 - 20 = 10 so b becomes 10
    a=a-b;  // now a becomes 30-10 = 20  ... so after swap a becomes 20
            // b becomes 10

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

int main()
{
    int a,b;
    printf(" Enter the values of a and b \n");
    scanf("%d%d",&a,&b);
    printf(" Before swapping a = %d and b = %d \n\n",a,b);
    swap_without_temp(a,b);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
 Enter the values of a and b
15 35
 Before swapping a = 15 and b = 35

 After swapping a = 35 and b = 15
```

1.c Compute the GCD of two integers and return the result to the calling function.

```c
#include<stdio.h>

int gcd_calc(int a,int b)
{
    if(b==0)
        return a;
    else
        return gcd_calc(b,a%b);
}

int main()
{
    int a,b;
    printf("Enter any two numbers (greater or equal to 0)\n\n");
    scanf("%d%d",&a,&b);
    int ans = gcd_calc(a,b);
    printf("The GCD of %d and %d is--- %d\n",a,b,ans);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter any two numbers (greater or equal to 0)

8 7
The GCD of 8 and 7 is--- 1

D:\compAssignment\assignment4>a.exe
Enter any two numbers (greater or equal to 0)

6 2
The GCD of 6 and 2 is--- 2
```

1.d Compute and returns the sum of n elements of an integer array.

```c
#include<stdio.h>

int sum_of_arr(int arr[],int n,int reqsize)
{
    int sum=0;
    for (int i = 0; i < reqsize; i++)
    {
        sum=sum + arr[i];
    }
    return sum;
}

int main()
{
    int n,reqsize;
    printf("Enter the size of array\n");
    scanf("%d",&n);
    int arr[n];
    printf("Enter the array elements one by one\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Enter the size upto which you want to add in the array\n");
    scanf("%d",&reqsize);
    if(reqsize>n)
        printf("Wrong input\n");
    else
    {
        int ans = sum_of_arr(arr,n,reqsize);
        printf("\nThe sum of array upto required size is --> %d",ans);
    }
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter the size of array
5
Enter the array elements one by one
1 3 2 5 4
Enter the size upto which you want to add in the array
3

The sum of array upto required size is --> 6
```

2. Write a C recursive function to find the factorial of an integer *N*.

```c
#include<stdio.h>

long int fact(long int n)
{
    if(n==0 || n==1)
        return 1;
    else
        return n*fact(n-1);
}
int main()
{
    long int ans,num;
    printf(" Enter any number greater than or equal to zero \n");
    scanf("%ld",&num);
    ans = fact(num);
    printf(" The factorial of %d is ---  %ld\n",num,ans);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
 Enter any number greater than or equal to zero
6
 The factorial of 6 is ---  720
```

3.Write C function for the following problem:

a. For a natural number find out its factors.

```c
#include<stdio.h>
void factor(int n)
{
    for(int i=1;i<=n;i++)
    {
        if(n%i==0)
            printf("%d ",i);
    }
}
int main()
{
    int num;
    printf("Enter any number\n");
    scanf("%d",&num);
    printf("The factors of %d are --->\n",num);
    factor(num);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter any number
6
The factors of 6 are --->
1 2 3 6
```

3.Write C function for the following problem:

b. For a range of numbers, say, 1 to N, find out the factors of each number and determine the one that has got a maximum number of factors.

```c
#include<stdio.h>
void factor(int n)
{
    int final_num=0;
    int count=0;
    int max_count=0;
    for(int i=1;i<=n;i++)
    {
        count=0;
        for(int j=1;j<=i;j++)
        {
            if(i%j==0)
            {
                count++;
            }
        }
        if(count>max_count)
        {
            max_count=count;
            final_num=i;
        }
    }
    printf("\n\nBetween 1 to %d,  %d has maximum number of factors which equals %d",n,final_num,max_count);
}

int main()
{
    int num;
    printf("Enter any number\n");
    scanf("%d",&num);
    factor(num);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter any number
1000



Between 1 to 1000,  840 has maximum number of factors which equals 32
```

4. Write a C function *reverse (s)* to reverse the string *s*, where s is an argument in the function *reverse (s)*.

```c
#include<stdio.h>
#include<string.h>
void reverse(char s[],int n)
{
    char c;
    for(int i=0;i<n/2;i++)
    {
        c=s[i];
        s[i]=s[n-i-1];
        s[n-i-1]=c;
    }
    printf("The string after reverse is -->\n");
    puts(s);

}

int main()
{
    static char s[100];
    printf("Enter the string\n");
    gets(s);
    int n=0;
    while(s[n]!='\0')
    {
        n++;
    }
    reverse(s,n);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter the string
algorithm
The string after reverse is -->
mhtirogla
```

5. Write a C function that takes input a two-dimensional array of integers and find the largest integer among them and return it to calling function.

```c
#include<stdio.h>

int max_ele(int r,int c,int arr[][c])
{
    int maxele = arr[0][0];
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            if(arr[i][j]>maxele)
            {
                maxele = arr[i][j];
            }
        }

    }
    return maxele;

}
int main()
{
    int r,c;
    printf("Enter number of rows and columns\n");
    scanf("%d%d",&r,&c);
    int arr[r][c];
    printf("Enter the elements of the 2D array\n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            scanf("%d",&arr[i][j]);
        }

    }
    int ans = max_ele(r,c,arr);
    printf("The largest element in the entire 2D array is --> %d\n",ans);
    return 0;
}
```

```
D:\compAssignment\assignment4>a.exe
Enter number of rows and columns
4 4
Enter the elements of the 2D array
1 4 5 7
8 9 6 3
11 32 31 20
15 12 10 21
The largest element in the entire 2D array is --> 32
```

6. Write a C function to multiple two two-dimensional matrices A and B and store the result in another matrix C.

```c
#include <stdio.h>

void getMatrixElements(int matrix[][10], int row, int column)
{
    printf("\nEnter elements: \n");
    for (int i = 0;i < row;i++)
    {
        for (int j = 0; j < column;j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}
```

```c
void multiplyMatrices(int first[][10],int second[][10],int result[][10],int r1,int c1,int r2,int c2)
{
   // Initializing elements of matrix mult to 0.
   for (int i = 0; i < r1; i++)
   {
      for (int j = 0; j < c2; j++)
      {
         result[i][j] = 0;
      }
   }
   for (int i = 0; i < r1;i++)
   {
      for (int j = 0; j < c2; j++)
      {
         for (int k = 0; k < c1; k++)
         {
            result[i][j] += first[i][k] * second[k][j];
         }
      }
   }
}
```

```c
int main()
{
    int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;
    printf("Enter rows and column for the first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for the second matrix: ");
    scanf("%d %d", &r2, &c2);
    if(c1 != r2)
    {
        printf("Error! The matrices can't be multiplied\n");
        return 0;
    }

   getMatrixElements(first, r1, c1);
   getMatrixElements(second, r2, c2);

   multiplyMatrices(first, second, result, r1, c1, r2, c2);
   printf("The resultant matrix after multiplication ----> \n\n")
    for(int i=0;i<r1;i++)
    {
        for(int j=0;j<c2;j++)
        {
            printf("%d ",result[i][j]);
        }
        printf("\n");
    }
}
```

```
D:\compAssignment\assignment4>a.exe
Enter rows and column for the first matrix: 3 3
Enter rows and column for the second matrix: 3 3

Enter elements:
1 2 3
3 4 2
3 2 1

Enter elements:
1 1 1
3 4 2
3 2 1
The resultant matrix after multiplication ---->

16 15 8
21 23 13
12 13 8
```