1. Write a program to input two matrices and do Matrix addition and subtraction

```c
#include <stdio.h>
//Matrix addition and subtraction
int main()
{

    int matrix1[3][3], matrix2[3][3];
    int result[3][3]; //for printing addition and
subtraction
    int i, j;

    printf("Enter matrix elements for first matrix: \n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            printf("Enter element at matrix1[%d][%d]", i+1,
j+1);
            scanf("%d", &matrix1[i][j]);
        }

    }
    printf("Enter matrix elements for Second matrix: \n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
```

```
            printf("Enter element at matrix2[%d][%d]", i+1,
j+1);
            scanf("%d", &matrix2[i][j]);
        }
    }
    //Printing First Matrix
    printf("First matrix:\n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            printf("%5d", matrix1[i][j]);
        }
        printf("\n");
    }

    //Printing Second Matrix
    printf("Second matrix:\n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            printf("%5d", matrix2[i][j]);
        }
        printf("\n");
    }

    //Matrix Addition
    printf("Matrix after Addition:\n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
```

```
        {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
            printf("%5d", result[i][j]);
        }
        printf("\n");
    }

    //Matrix Subtraction
    printf("Matrix after Subtraction:\n");
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            result[i][j] = matrix1[i][j] - matrix2[i][j];
            printf("%5d", result[i][j]);
        }
        printf("\n");
    }
     return 0;
}
```

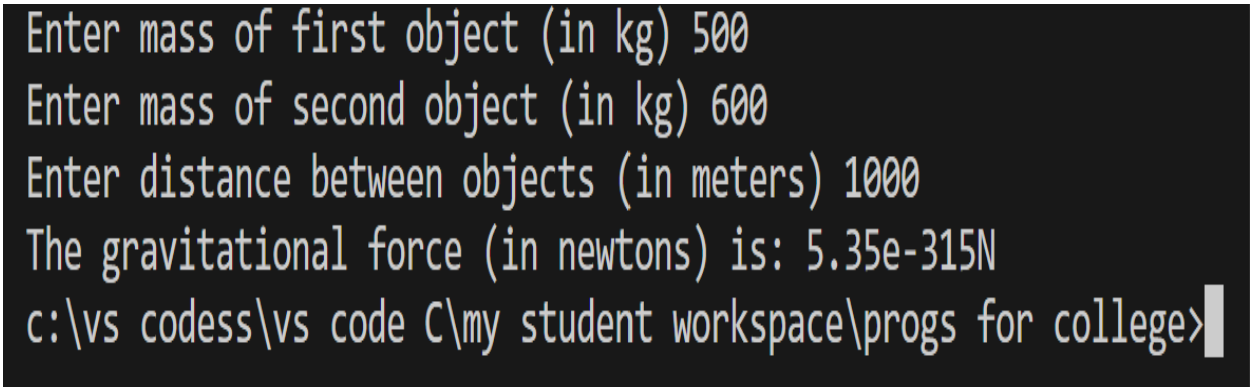
Output:

```
Enter matrix elements for first matrix:
Enter element at matrix1[1][1]1
Enter element at matrix1[1][2]2
Enter element at matrix1[1][3]3
Enter element at matrix1[2][1]4
Enter element at matrix1[2][2]5
Enter element at matrix1[2][3]6
Enter element at matrix1[3][1]7
Enter element at matrix1[3][2]8
Enter element at matrix1[3][3]9
Enter matrix elements for Second matrix:
Enter element at matrix2[1][1]2
Enter element at matrix2[1][2]4
Enter element at matrix2[1][3]6
Enter element at matrix2[2][1]8
Enter element at matrix2[2][2]10
Enter element at matrix2[2][3]12
Enter element at matrix2[3][1]14
Enter element at matrix2[3][2]16
Enter element at matrix2[3][3]18
First matrix:
     1     2     3
     4     5     6
     7     8     9
Second matrix:
     2     4     6
     8    10    12
    14    16    18
Matrix after Addition:
     3     6     9
    12    15    18
    21    24    27
Matrix after Subtraction:
    -1    -2    -3
    -4    -5    -6
    -7    -8    -9
```

2. Write a program to calculate the Gravitational Force between two objects, taking G as $6.67 * 10^{-11}$.

```c
#include <stdio.h>

#define G 6.67e-11


double gravitational_force(double m1, double m2, double d)

{

    return G*m1*m2/(d*d);

}

int main()

{

    double m1, m2, d, force;

    printf("Enter mass of first object (in kg) ");

    scanf("%lf", &m1);

    printf("Enter mass of second object (in kg) ");

    scanf("%lf", &m2);

    printf("Enter distance between objects (in meters) ");

    scanf("%lf", &d);

    force = gravitational_force(m1,m2,d);

    printf("The gravitational force (in newtons) is: %.2eN");

    return 0;

   }
```

Output:


Enter mass of first object (in kg) 500
Enter mass of second object (in kg) 600
Enter distance between objects (in meters) 1000
The gravitational force (in newtons) is: 5.35e-315N
c:\vs codess\vs code C\my student workspace\progs for college>

3. Write a C program to check if a given number remains the same even after reversing its digits(without using string functions). These types of numbers are also called palindrome numbers. For example, 121, 1331, 2468642 are these types of numbers, but 123, 145, 9876 are not.

```c
#include <stdio.h>

int main()
{

     int n, reversed = 0, remainder;
     printf("This program is to check whether a number is palindrome or not \nEnter the number you want to check ");
     scanf("%d", &n);
     int original = n; // because we'll do manupulations on n so we need original number saved

     while (n != 0)
     {
          remainder = n % 10;                    // to get the last digit
          reversed = reversed * 10 + remainder; // to keep appending digits
          n = n / 10;
     }
     if (reversed == original)
     {
          printf("Yes, %d is a palindrome number", original);
```

```
    }
    else
    {
        printf("No, %d is not a palindrome number", original);
    }
    return 0;
}
```

Output:

```
This program is to check whether a number is palindrome or not
Enter the number you want to check 2468642
Yes, 2468642 is a palindrome number
c:\vs codess\vs code C\my student workspace\progs for college>
```

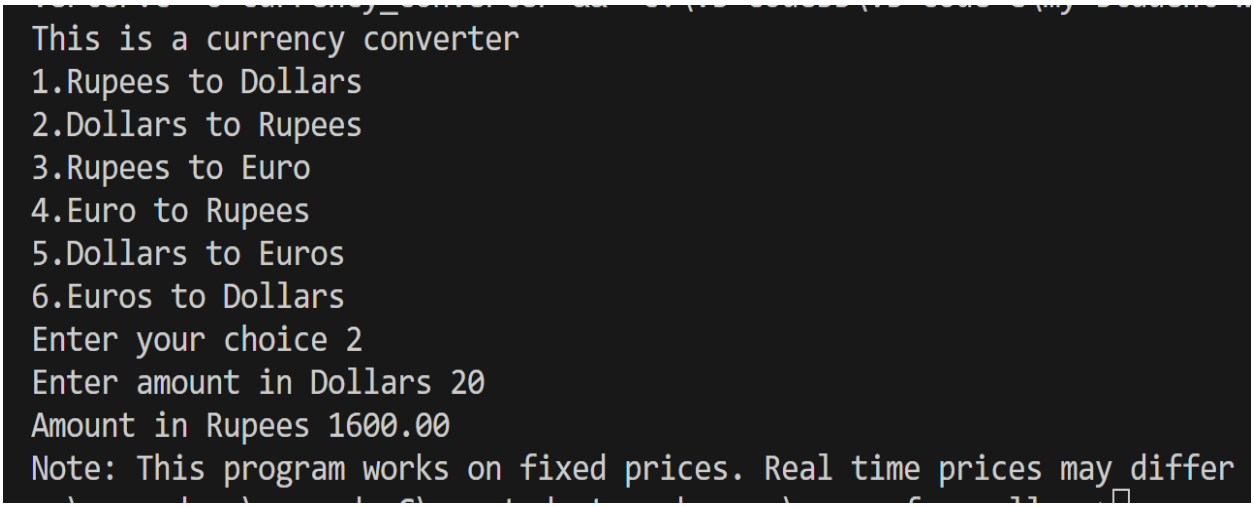
4. Write a program to make a currency converter, that can convert minimum 3 currencies into one another(say USD, EURO & INR). Keeping prices fixed and ignoring real time change.

```c
#include <stdio.h>

//1USD = 80INR, 1EURO = 85INR, 1EURO= 1.06USD

int main()
{
    int choice;
    float rupees, dollars, euros;
    printf("This is a currency converter\n");
    printf("1.Rupees to Dollars \n");
    printf("2.Dollars to Rupees \n");
    printf("3.Rupees to Euro \n");
    printf("4.Euro to Rupees \n");
    printf("5.Dollars to Euros \n");
    printf("6.Euros to Dollars \nEnter your choice ");
    scanf("%d", &choice);

    switch (choice)
    {
    case 1: //Rupees to Dollars
        printf("Enter amount in Rupees ");
        scanf("%f", &rupees);
        dollars = rupees/80;
        printf("Amount in Dollars %.2f $", dollars);
```

```
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    case 2: //Dollars to Rupees
        printf("Enter amount in Dollars ");
        scanf("%f", &dollars);
        rupees = dollars*80;
        printf("Amount in Rupees %.2f ", rupees);
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    case 3: //Rupees to Euro
        printf("Enter amount in Rupees ");
        scanf("%f", &rupees);
        euros = rupees/85;
        printf("Amount in euros %.2f ", euros);
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    case 4: //Euro to Rupees
        printf("Enter amount in Euros ");
        scanf("%f", &euros);
        rupees = euros*85;
        printf("Amount in Rupees %.2f ", rupees);
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    case 5: //Dollars to Euros
        printf("Enter amount in Dollars ");
        scanf("%f", &dollars);
```

```
        euros = dollars/1.06;
        printf("Amount in Euros %.2f ", euros);
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    case 6: //Euros to Dollars
        printf("Enter amount in euros ");
        scanf("%f", &euros);
        dollars = euros*1.06;
        printf("Amount in Dollars %.2f $", dollars);
        printf("\nNote: This program works on fixed prices.
Real time prices may differ");
        break;

    default:
        printf("Wrong input");
        break;
    }
    return 0;
}
```

Output:



5. Write a program that takes user input n and prints numbers which are the cube roots of the product of their proper divisors upto n. For example 24 = 2,3,4,6,8,12, their product is 13,824 whose cube root is again 24.

```c
#include <stdio.h>

int productOfDivisors(int num)
{
    int product = 1;
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0) {
            product *= i;
        }
    }
    return product;
}

int main() {
    int n;

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

    printf("Numbers up to %d whose cube roots are equal to the product of their proper divisors:\n", n);

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

```
        int cube = i * i * i;
        int divisorProduct = productOfDivisors(i);

        if (cube == divisorProduct)
        {
            printf("%5d ", i);
        }
    }
    printf("\n");

    return 0;
}
```

Output:

```
Enter a value for n: 100
Numbers up to 100 whose cube roots are equal to the product of their proper divisors:
    1    24    30    40    42    54    56    66    70    78    88
```

6. Write a program that asks a user input but it keeps asking input until he enters 0. Print the product of all user input numbers and number of times user did input a number.

```c
#include <stdio.h>

int main()

{
    int n, product = 1, count = 0;
    printf("Give me a number and I will end this program
");
    scanf("%d", &n);
    product = product*n;

    while (n!=0)
    {
        printf("Crap, try again ");
        scanf("%d", &n);
        if (n!=0)
        {
            product = product*n;
            count +=1;
        }
    }
    printf("Damn, you smarty pants you entered
zero\nAnyways, here is your product %d\nAnd no. of times
```

```
you were forced to input number is %d :)", product,
count);
     return 0;
}
```

Output:

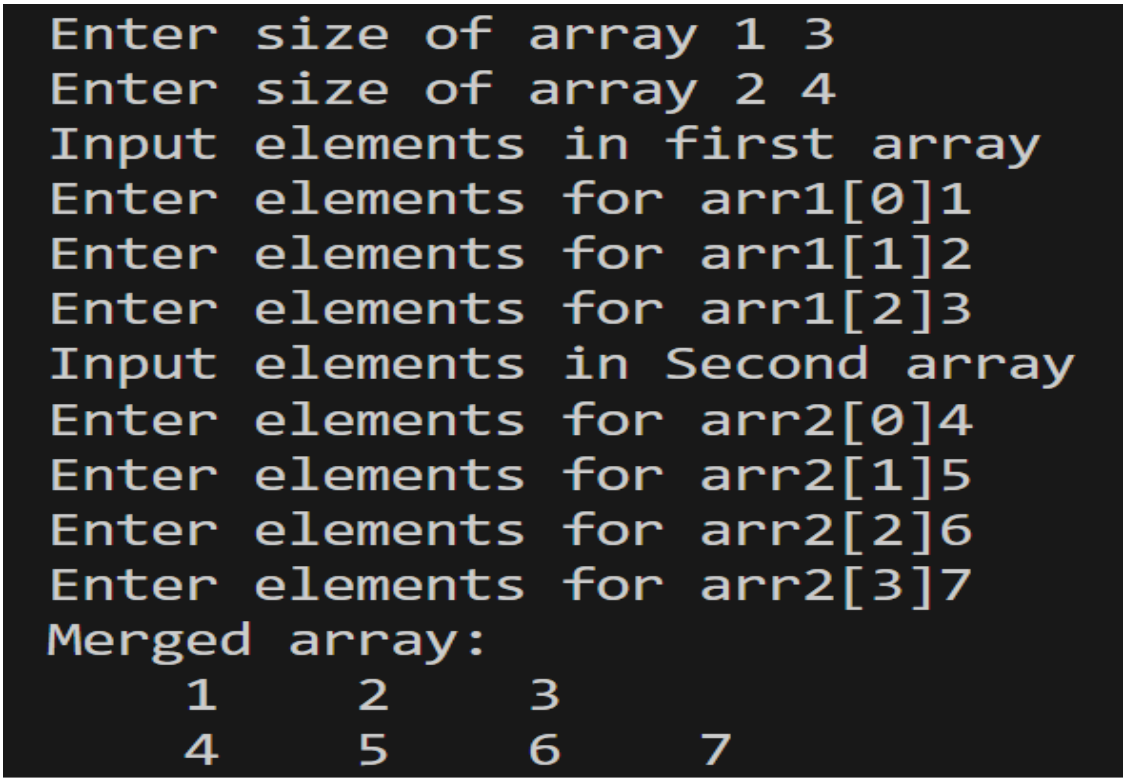
```
Give me a number and I will end this program 25
Crap, try again 24
Crap, try again 22
Crap, try again 23
Crap, try again 17
Crap, try again 0
Damn, you smarty pants you entered zero
Anyways, here is your product 5161200
And no. of times you were forced to input number is 4 :)
```

7. Write a program to merge two arrays into a third array.

```c
#include <stdio.h>

int main()
{
    int size_one, size_two; //size of array
    int count = 0, tout = 0;
    printf("Enter size of array 1 ");
    scanf("%d", &size_one);
    printf("Enter size of array 2 ");
    scanf("%d", &size_two);

    int i, j, k, m, n;
    int tot_size; //size of final array
    int arr1[size_one], arr2[size_two];

    //First array input loop
    printf("Input elements in first array\n");
    for (i = 0; i < size_one; i++)
    {
        printf("Enter elements for arr1[%d]", i);
        scanf("%d", &arr1[i]);
    }

    //Second array input loop
    printf("Input elements in Second array\n");
    for (k = 0; k < size_two; k++)
    {
        printf("Enter elements for arr2[%d]", k);
        scanf("%d", &arr2[k]);
    }

    //Merging in final array
    tot_size = size_one + size_two;
    int finarr[tot_size]; //final array declare

    for (j = 0; j < size_one; j++)
```

```
    {
        finarr[j] = arr1[j]; //merging first into final
    }

    for (m = size_one; m < tot_size; m++)
    {
        finarr[m] = arr2[m-size_one]; //merging second into final
    }
    int pout = 0;
    printf("Merged array:\n");
    for (n = 0; n < tot_size; n++)
    {
        printf("%5d", finarr[n]);

        if (n==size_one-1)
        {
            printf("\n"); //to print in new line for elements for second
array that are now merged
        }

    }

    return 0;

}
```

Output:


Enter size of array 1 3
Enter size of array 2 4
Input elements in first array
Enter elements for arr1[0]1
Enter elements for arr1[1]2
Enter elements for arr1[2]3
Input elements in Second array
Enter elements for arr2[0]4
Enter elements for arr2[1]5
Enter elements for arr2[2]6
Enter elements for arr2[3]7
Merged array:
1 2 3
4 5 6 7

8. Create a structure of Student. Input details like roll number, name, and and marks of 3 subjects(phy, chem, math). Show the name, marks of 3 subjects and also the total in a bit organized way.

```c
#include <stdio.h>
#include <string.h>
struct student
{
    char name[30];
    int phy_marks;
    int mat_marks;
    int che_marks;
    int roll;
};

int total;
int main()
{

    int i, j, n; // n for number of students
    // i & j for counter

    printf("Hii, This program is to input student details, marks and get their total\nEnter the no. of students you want to input details of");
    scanf("%d", &n);

    struct student st[n]; // making array of student datatype that contains n students

```

```
    for (i = 0; i < n; i++)
    {
        printf("Enter name of student[%d] ", i + 1);
        scanf("%s", st[i].name);
        printf("Enter roll number of student[%d] ", i + 1);
        scanf("%d", &st[i].roll);
        printf("Enter marks of physics of student[%d] ", i + 1);
        scanf("%d", &st[i].phy_marks);
        printf("Enter marks of mathematics of student[%d] ", i + 1);
        scanf("%d", &st[i].mat_marks);
        printf("Enter marks of chemistry of student[%d] ", i + 1);
        scanf("%d", &st[i].che_marks);
    }
    printf("Details of each student below:\n");
    printf("Name \t\tMarks\n");
    for (i = 0; i < n; i++)
    {
        printf("%s \t\n", st[i].name);
        printf(" \t\t%d(Physics)\n", st[i].phy_marks);
        printf(" \t\t%d(Mathematics)\n", st[i].mat_marks);
        printf(" \t\t%d(Science)\n", st[i].che_marks);
        total = st[i].phy_marks + st[i].mat_marks + st[i].che_marks;
        printf("Total \t\t%d\n\n", total);
    }

    return 0;
}
```

Output:

```
Hii, This program is to input student details, marks and get their total
Enter the no. of students you want to input details of2
Enter name of student[1] Harry
Enter roll number of student[1] 21
Enter marks of physics of student[1] 20
Enter marks of mathematics of student[1] 30
Enter marks of chemistry of student[1] 40
Enter name of student[2] Brijesh
Enter roll number of student[2] 22
Enter marks of physics of student[2] 80
Enter marks of mathematics of student[2] 90
Enter marks of chemistry of student[2] 70
Details of each student below:
Name            Marks
Harry
                20(Physics)
                30(Mathematics)
                40(Science)
Total           90

Brijesh
                80(Physics)
                90(Mathematics)
                70(Science)
Total           240
```

9. Write a program to know the day of a particular date.

```c
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>

int main()

{
    int day, month, year;
    printf("Enter date in DD MM YYYY format ");
    scanf("%d %d %d", &day, &month, &year);

    int daysinmonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    if(month==2)
    {
        if((year % 4 ==0 && year % 100 != 0) || (year % 400 == 0))
        {
            daysinmonth[2] = 29;
        }
    }
    char* weekdays[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"}; //0 = Sunday, 6 = Saturday

    if (day<1 || day > 31 || month<1 || month > 12 || year<1950 || year > 9999)
    {
        printf("Invalid input ");
    }
    else{
```

```c
        int totaldays = 0;
        for (int y = 1950; y < year; y++)
        {
            totaldays += 365;
            if((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))
        {
            totaldays += 1;
        }
        }
        for (int m = 1; m < month; m++)
        {
            totaldays += daysinmonth[m];
        }

        totaldays += day - 1; // subtracting 1 because days start with 1 so it doesn't
add 2 for 2nd of Jan, 1950
        int weekdaysindex = totaldays % 7;
        // printf("weekdaysindex = %d", weekdaysindex);
        printf("The day is %s", weekdays[weekdaysindex]);

    }
    return 0;
}
```

Output:

```
Enter date in DD MM YYYY format 17 08 2023
The day is Thursday
c:\vs codess\vs code C\my student workspace\dummy progs f
dum_dayofweek.c -o dum_dayofweek && "c:\vs codess\vs code
Enter date in DD MM YYYY format 30 09 2023
The day is Saturday
```