OPERATING SYSTEM

ASSESMENT 1

By: Sparsh Arya

Registration number:17BEC0656

Slot: L7(Lab)

Date of submission: 31st July 2018.

Total number of pages: 37

Question 1.

Menu driven C program to perform basic arithmetic operations. (Hint: Get corresponding operators and values as input).

Algorithm:

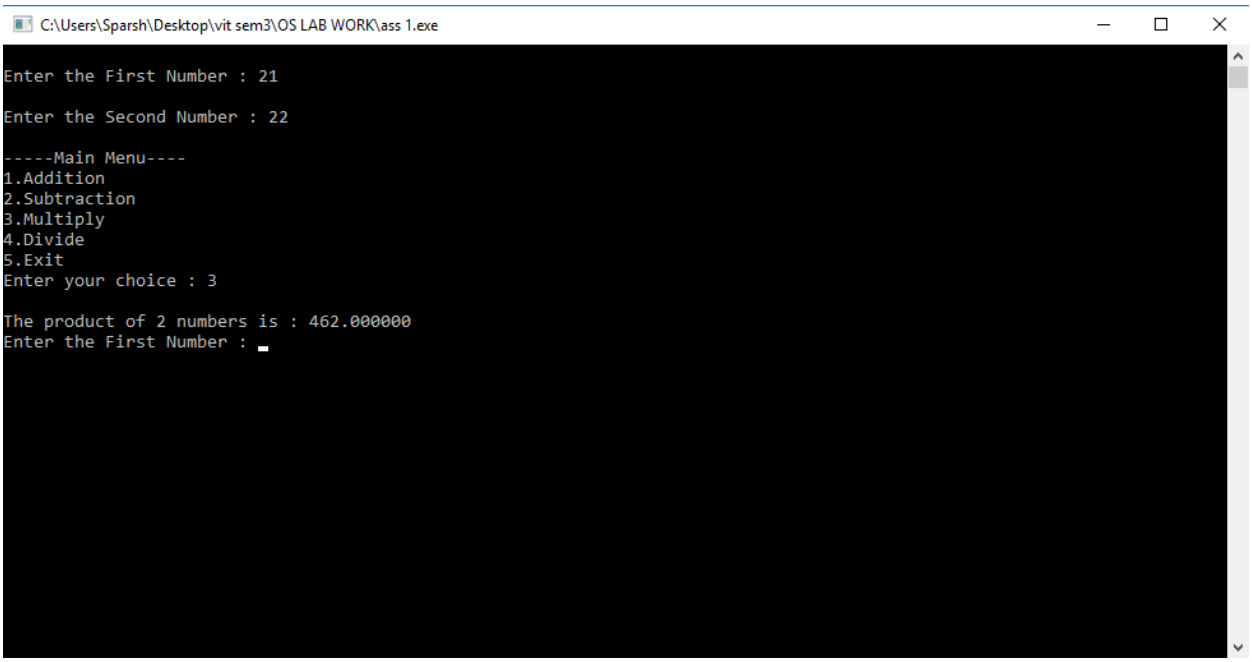
- Create a do while loop with the condition till when the user wants to run the program and not exit it.
- Take 2 floating inputs and store it in num1 and num2.
- Then initialize certain numbers for desired operations to be performed at those numbers.
- Declare:
    - 1- addition.
    - 2-subtraction.
    - 3-multiplication.
    - 4-divide.
    - 5-exit (Run this operation to get out of the loop in order to stop executing any further commands).
- Make use of switch iterative statements to perform various operations like addition, subtraction, multiplication etc.
- In case of a number entered except the above shown numbers, include a default statement saying that the user has entered the wrong choice.
- Run out of the loop.

Source code:

```
#include<stdio.h>
int main()
{
  float num1,num2,ans;
  int opt;
 //taking user input
        do
              {
                     printf("\nEnter the First Number : ");
                    scanf("%f",&num1);
                    printf("\nEnter the Second Number : ");
                    scanf("%f",&num2);

                    //Displaying menu
                    printf("\n-----Main Menu----\n1.Addition");
                    printf("\n2.Subtraction\n3.Multiply\n4.Divide\n5.Exit");
                    printf("\nEnter your choice : ");
                    scanf("%d",&opt);

                    switch(opt)
                    {
                           case 1:
```

```
                    ans = num1+num2;
                printf("\nThe addition of 2 numbers is : %f",ans);
                 break;
            case 2:
                ans = num1-num2;
                printf("\nThe differnce of 2 numbers is : %f",ans);
                break;
            case 3:
                ans = num1*num2;
                printf("\nThe product of 2 numbers is : %f",ans);
                break;
            case 4:
                ans = num1/num2;
                printf("\nThe division of 2 numbers is : %f",ans);
                break;
            case 5:
                break;
            default:  //error message for wrong choice
                    printf("\nYou Entered Wrong Choice\n");
                    break;
        }
  }while(opt!=5);
 return 0;
}
```

Output:



Question 2.

Menu driven C program to swap three integers using and without using temp variable.

Part1

Algorithm for swapping 3 integers using a temp variable:

- Take input for 3 numbers and store it as firstnumber, secondnumber and thirdnumber (all should be of int datatype).
- Store another variable as temp variable (int datatype).
- Use below algorithm:

- Temp=firstnumber.
- Firstnumber=secondnumber.
- Secondnumber=thirdnumber.
- Thirdnumber=temp.

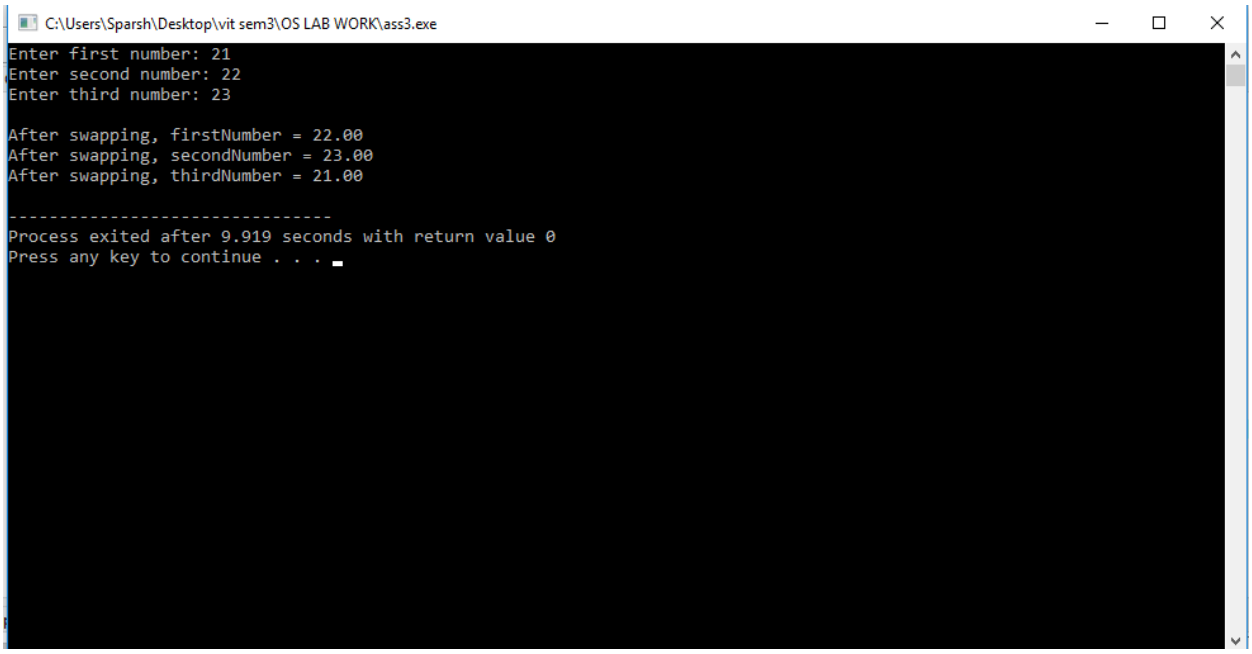
- Display all three interchanged variables.

Source code:

```
#include <stdio.h>
int main()
{
    int firstNumber, secondNumber, thirdNumber, temp;

    printf("Enter first number: ");
    scanf("%lf", &firstNumber);

    printf("Enter second number: ");
    scanf("%lf",&secondNumber);

     printf("Enter third number: ");
    scanf("%lf", &thirdNumber);

    // Value of firstNumber is assigned to temp
    temp = firstNumber;
```

```
    // Value of secondNumber is assigned to firstNumber
    firstNumber = secondNumber;

    // Value of third number  is assigned to secondNumber
    secondNumber = thirdNumber;

    // Value of temp; is assigned to thirdNumber
     thirdNumber=temp ;



    printf("\nAfter swapping, firstNumber = %.2lf\n", firstNumber);
    printf("After swapping, secondNumber = %.2lf", secondNumber);
    printf("\nAfter swapping, thirdNumber = %.2lf\n", thirdNumber);

    return 0;
}
```

Output



Part 2: Without using temp variable.

Algorithm for swapping 3 numbers using a temp variable:

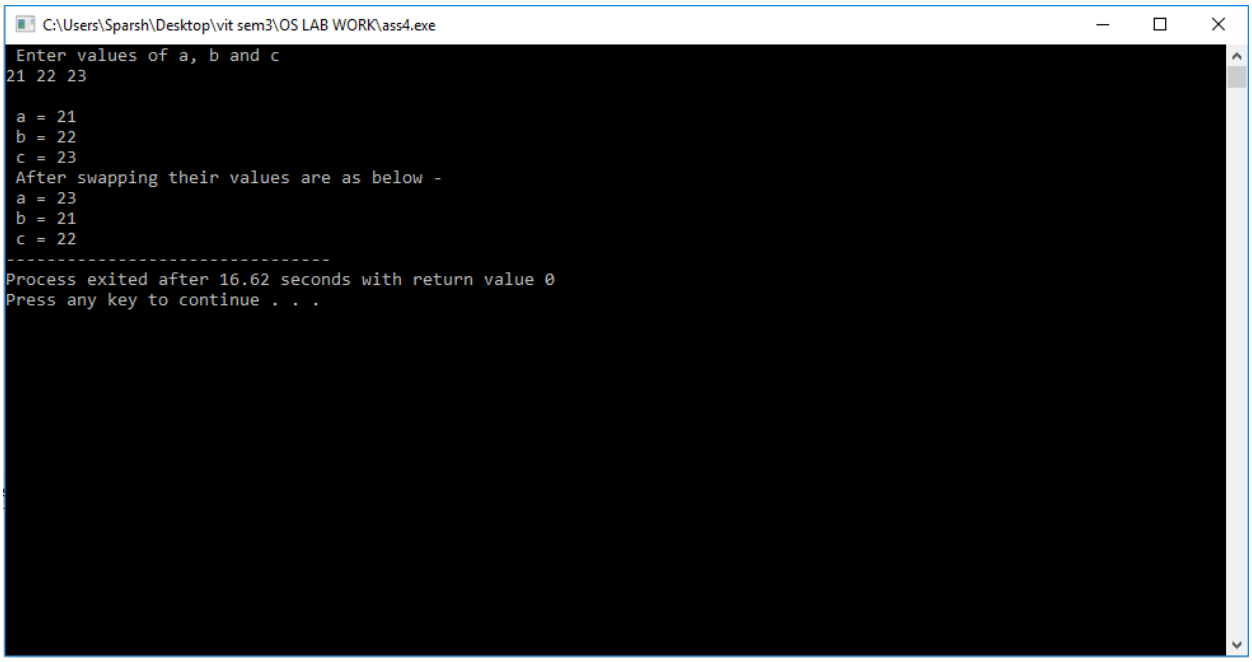
- Take input for 3 integers and store the values as integer datatype (in form of a, b and c).
- Perform the following operation on them.
- a=a+b+c;
- b=a-b-c;
- c=a-b-c;
- a=a-b-c;
- Display the values of a, b and c

Source code:

```
#include<stdio.h>


int main()
{
    int a,b,c;
    printf(" Enter values of a, b and c \n");
    scanf("%d %d %d",&a,&b,&c);
    printf("\n a = %d",a);
    printf("\n b = %d",b);
    printf("\n c = %d",c);
    a=a+b+c;
    b=a-b-c;
    c=a-b-c;
    a=a-b-c;



        printf("\n After swapping their values are as below -");
        printf("\n a = %d",a);
    printf("\n b = %d",b);
    printf("\n c = %d",c);
    return 0;
```

}

Output:



Question 3.

Menu driven C program to read N integers and perform sorting and searching.

Algorithm for sorting:

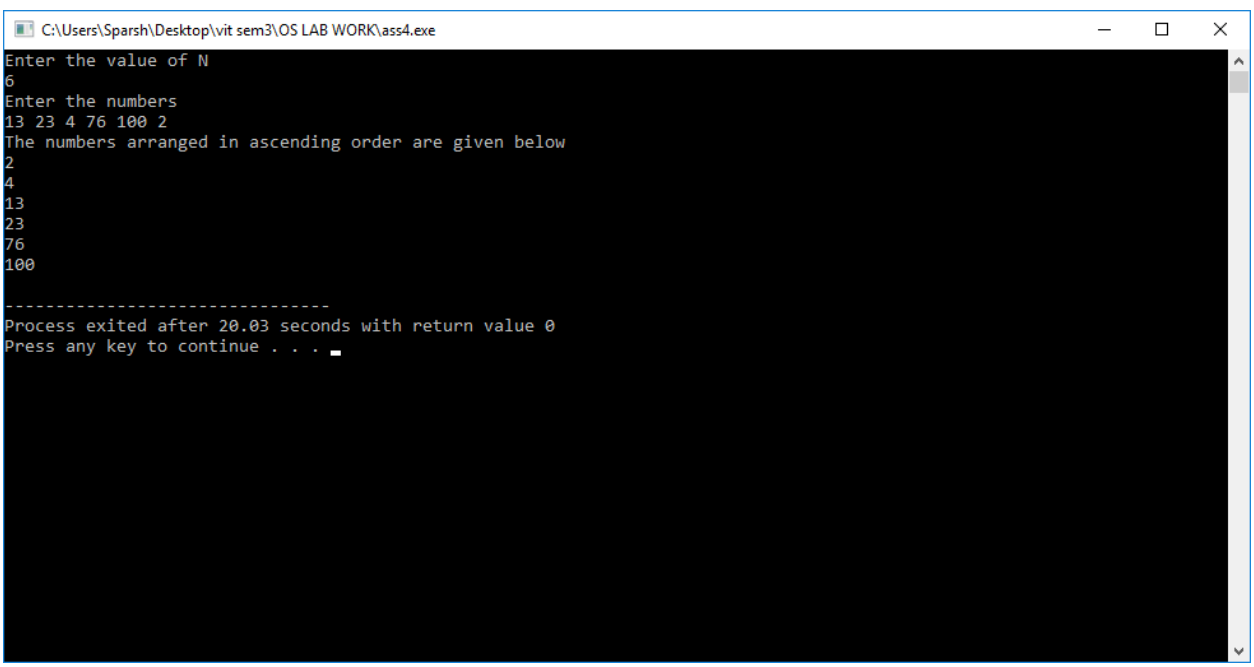
- Take input for the number of numbers in the array.
- Take the array input.
- Run 2 simultaneous for loops to compare the values of the numbers in the array, if the first number is greater than the second number than reverse the 2 numbers.
- Finally, display the array.

Source code:

```
#include <stdio.h>
   int main()
  {

      int i, j, a, n, number[30];
      printf("Enter the value of N \n");
      scanf("%d", &n);

      printf("Enter the numbers \n");
      for (i = 0; i < n; ++i)
         scanf("%d", &number[i]);

      for (i = 0; i < n; ++i)
      {

         for (j = i + 1; j < n; ++j)
         {

            if (number[i] > number[j])
            {

               a =  number[i];
               number[i] = number[j];
```

```
            number[j] = a;

        }

    }

}

    printf("The numbers arranged in ascending order are given below \n");
    for (i = 0; i < n; ++i)
        printf("%d\n", number[i]);
return 0;
  }
```

Output:



Part 2: Searching an element in the array

Algorithm:

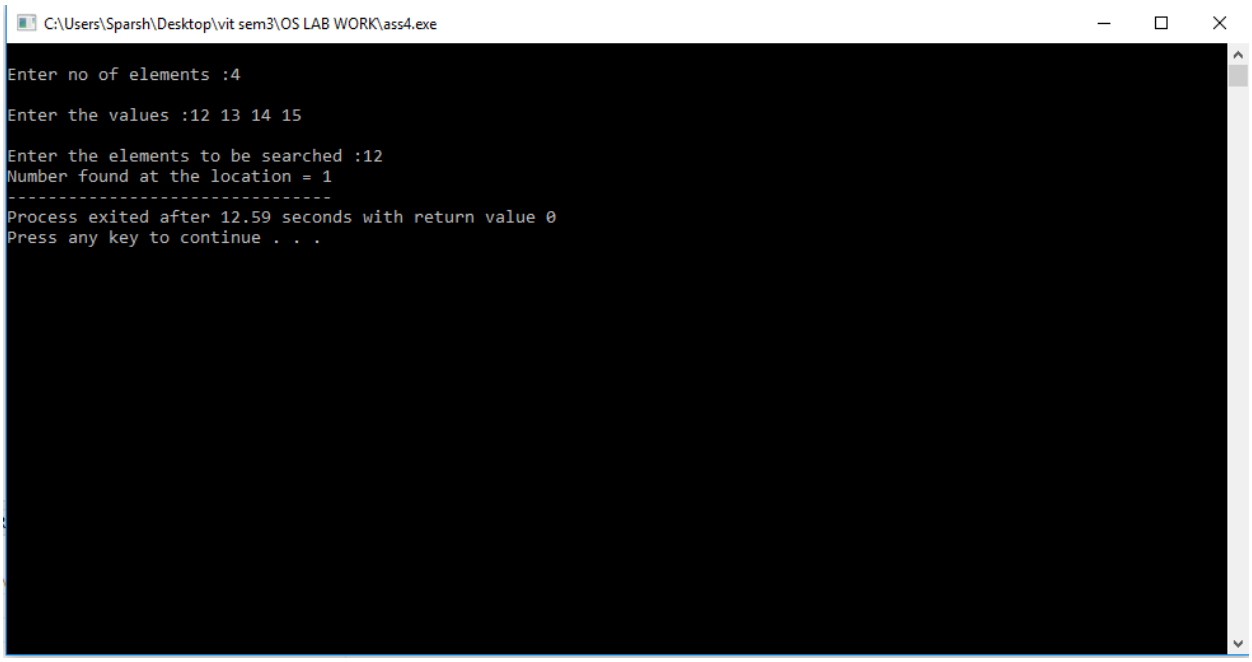
- Input all the elements into an array.
- Use a for loop to access every element of the array and compare if it is equal to the search element.
- If such a element is found then display the value of the index.
- If no so such element is found then display a message saying that the element doesn't exist in the array.

Source code:

```
#include <stdio.h>


int main() {
  int a[30], ele, num, i;


  printf("\nEnter no of elements :");
  scanf("%d", &num);


  printf("\nEnter the values :");
  for (i = 0; i < num; i++) {
```

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

  //Read the element to be searched
  printf("\nEnter the elements to be searched :");
  scanf("%d", &ele);

  //Search starts from the zeroth location
  i = 0;
  while (i < num && ele != a[i]) {
    i++;
  }

  //If i < num then Match found
  if (i < num) {
    printf("Number found at the location = %d", i + 1);
  } else {
    printf("Number not found");
  }

  return (0);
}
```

Output:



Question 4.

Menu driven C program to perform matrix addition, subtraction and multiplication operations.

Algorithm:

- Make use of switch statement and take input from the user as to which operation is to be performed.
- Take the 2 input matrices.
- Perform simple matrix addition, subtraction and multiplication with the help of functions.
- These functions are called from the switch statement.
- For matrix addition, use a for loop for adding the subsequent elements.

- For matrix subtraction, use a for loop for subtracting subsequent elements.
- For matrix multiplication, multiply every row element of matrix 1 with the column elements of matrix 2.
- Finally display the result through switch statement.

Source code:

```
# include<stdio.h>
    void display(int [][3]);
    void main()
    {
      int c;
      void func1();
      void func2();
      void func3();
      void func4();
      void func5();
      clrscr();
        printf("\n- : Matrix Manipulation Functions (for 3 X 3 Matrix) : -");
      printf("\n-------------------------------------");
      printf("\n Matrix Addition          : 1");
      printf("\n Matrix Subtraction       : 2");
      printf("\n Matrix Multiplication    : 3");
```

```
printf("\n Find Transpose Matrix      : 4");
printf("\n Matrix is Symmetric or not : 6");
printf("\n Enter Your Choice          : ");
scanf("%d",&c);
switch(c)
{
   case 1:
      func1();
      break;
   case 2:
      func2();
      break;
   case 3:
      func3();
      break;
   case 4:
      func4();
      break;
   case 5:
      func5();
      break;
   default:
      printf("\nInvalid Choice");
}
```

```
    getch();
}


void func1()
{
    int x[3][3],y[3][3],z[3][3];
    void getmatrix(int [][3]);
    void addition(int [][3],int [][3],int [][3]);
    clrscr();
    getmatrix(x);
    getmatrix(y);
    addition(x,y,z);
    printf("\n - : Matrix 1: - \n");
    display(x);
    printf("\n - : Matrix 2: - \n");
    display(y);
    printf("\n - : Matrix Addition (Result): - \n");
    display(z);
}
void getmatrix(int t[][3])
{
```

```
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("Enter element [%d][%d] : ",i,j);
            scanf("%d",&t[i][j]);
        }
    }
}
void addition(int p[][3],int q[][3],int r[][3])
{    int i,j;
    for(i=0;i<3;i++)
    {     for(j=0;j<3;j++)
            r[i][j]=p[i][j]+q[i][j];
    }
}
void func2()
{
    int x[3][3],y[3][3],z[3][3];
    void getmatrix(int [][3]);
    void subtraction(int [][3],int [][3],int [][3]);
    clrscr();
    getmatrix(x);
```

```
    getmatrix(y);
    subtraction(x,y,z);
    printf("\n - : Matrix 1: - \n");
    display(x);
    printf("\n - : Matrix 2: - \n");
    display(y);
    printf("\n - : Matrix Subtraction (Result): - \n");
    display(z);
}
void subtraction(int p[3][3],int q[3][3],int r[3][3])
{
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            r[i][j]=p[i][j]-q[i][j];
    }
}
void func3()
{
    int x[3][3],y[3][3],z[3][3];
    void getmatrix(int [][3]);
    void multiplication(int [][3],int [][3],int [][3]);
    clrscr();
```

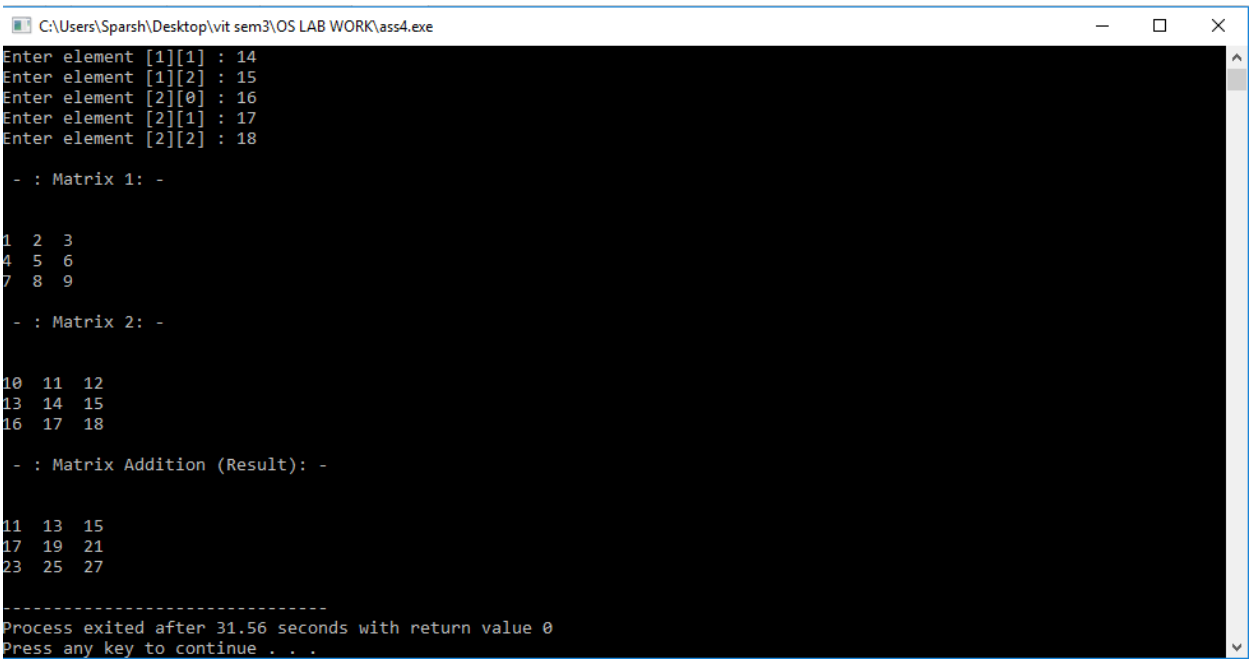
```
        getmatrix(x);
        getmatrix(y);
        multiplication(x,y,z);
        printf("\n - : Matrix 1: - \n");
        display(x);
        printf("\n - : Matrix 2: - \n");
        display(y);
        printf("\n - : Matrix Multiplication (Result): - \n");
        display(z);
    }
    void multiplication(int p[][3],int q[3][3],int r[3][3])
    {
        int i,j,k;
        for(i=0;i<3;i++)
//condition i< total row of matrix1
        {
            for(j=0;j<3;j++)
//condition i< total col of matrix1 or//condition i< total row of matrix2
{
                r[i][j]=0;
                for(k=0;k<3;k++) //condition i< total col of matrix2
                    r[i][j]=r[i][j]+(p[i][j]*q[j][k]);
            }
        }
```

```
}
void func4()
{
   int x[3][3],y[3][3];
   void getmatrix(int [][3]);
   void transpose(int [][3],int [][3]);
   clrscr();
   getmatrix(x);
   transpose(x,y);
   printf("\n - : Matrix 1: - \n");
   display(x);
   printf("\n - : Transpose Matrix : - \n");
   display(y);
}
void transpose(int p[][3],int q[][3])
{
   int i,j;
   for(i=0;i<3;i++)
   {
      for(j=0;j<3;j++)
         q[i][j]=p[j][i];
   }
}
void func5()
```

```
{
    int x[3][3],y[3][3];
    void getmatrix(int [][3]);
    void transpose(int [][3],int [][3]);
    int symmetric(int [][3],int [][3]);
    clrscr();
    getmatrix(x);
    transpose(x,y);
    if(symmetric(x,y)==1)
        printf("\nMatrix is Symmetric");
    else
        printf("\nMatrix is Not Symmetric");
}
int symmetric(int p[][3],int q[][3])
{
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(q[i][j]!=p[i][j])
                return 0;
        }
    }
```

```
    return 1;
}
void display(int m[][3])
{
    int i,j;
    printf("\n\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            printf("%d  ",m[i][j]);
        printf("\n");
    }
}
```

Output



Question 5.

C Program to print Fibonacci series.

Algorithm:

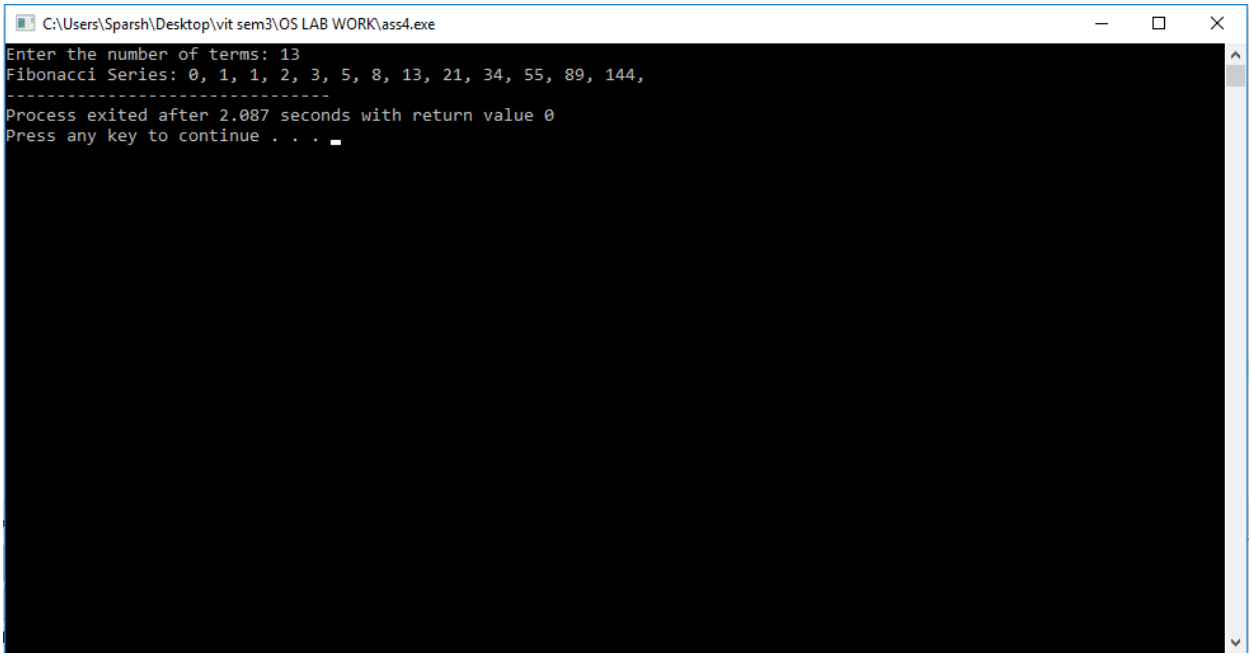
- Ask the user to input the term till which sequence is required.
- Initialize the first and the second term of the Fibonacci sequence as 0 and 1.
- Use a for loop to add new value to number by adding its previous 2 numbers
- Display the terms until till the value given at runtime.

Source code:

```
#include <stdio.h>

int main()

{
```

```
    int i, n, t1 = 0, t2 = 1, nextTerm;

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

    printf("Fibonacci Series: ");

    for (i = 1; i <= n; ++i)
    {
        printf("%d, ", t1);
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
    }
    return 0;
}
```

Output



Question 6.

C Program to find factorial of a given number.

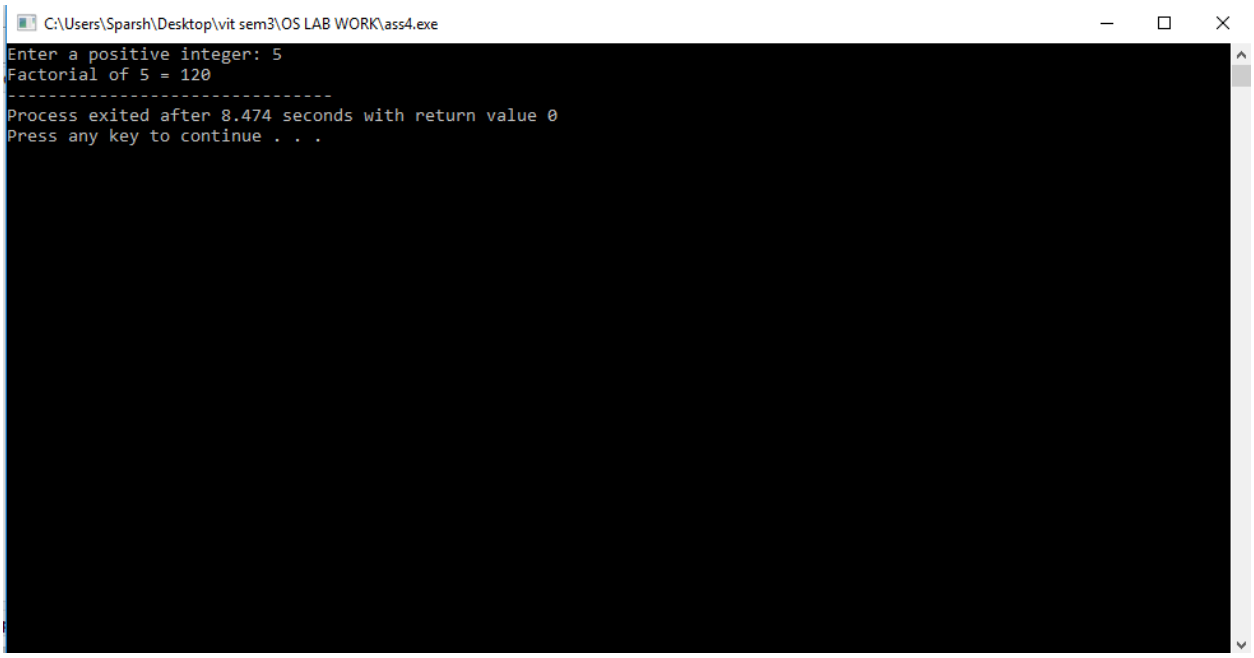
Algorithm:

- Take input for a positive number.
- Using a for loop, multiply all those numbers from 1 to the given number.
- In every cycle, store the value.
- After the loop breaks, display the value.

Source code:

```
#include <iostream>
using namespace std;

int main()
{
    unsigned int n;
    unsigned long long factorial = 1;

    cout << "Enter a positive integer: ";
    cin >> n;

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

    cout << "Factorial of " << n << " = " << factorial;
    return 0;
}
```

Output:



Question 7.

C Program to print prime number from 1 to n.

Algorithm:

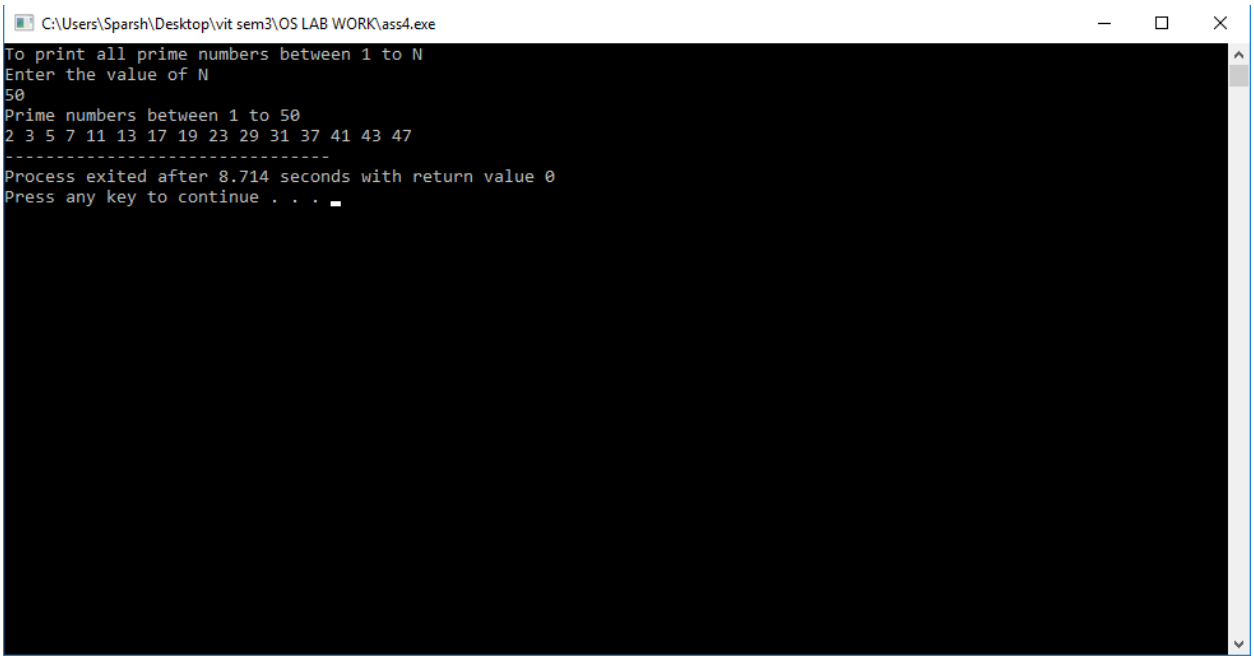
- Take input until which number u want to display prime numbers.
- Use a for loop and keep incrementing the number from to the desired number.
- In every for-loop cycle, create another loop to check whether the number has no factor except itself and 1.
- If the condition is satisfied then display the number.
- Else move ahead in the cycle by incrementing the number.

Source code:

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

int main()
{

   int N, i, j, isPrime, n;

   printf("To print all prime numbers between 1 to N\n");
   printf("Enter the value of N\n");
   scanf("%d",&N);

   /* For every number between 2 to N, check
   whether it is prime number or not */
   printf("Prime numbers between %d to %d\n", 1, N);

   for(i = 2; i <= N; i++){
      isPrime = 0;
      /* Check whether i is prime or not */
      for(j = 2; j <= i/2; j++){
         /* Check If any number between 2 to i/2 divides I
          completely If yes the i cannot be prime number */
         if(i % j == 0){
```

```
            isPrime = 1;
            break;
        }
    }

    if(isPrime==0 && N!= 1)
        printf("%d ",i);
  }

  return 0;
}
```

Output



Question 8.

C Program to find sum of first n natural numbers.

Algorithm:

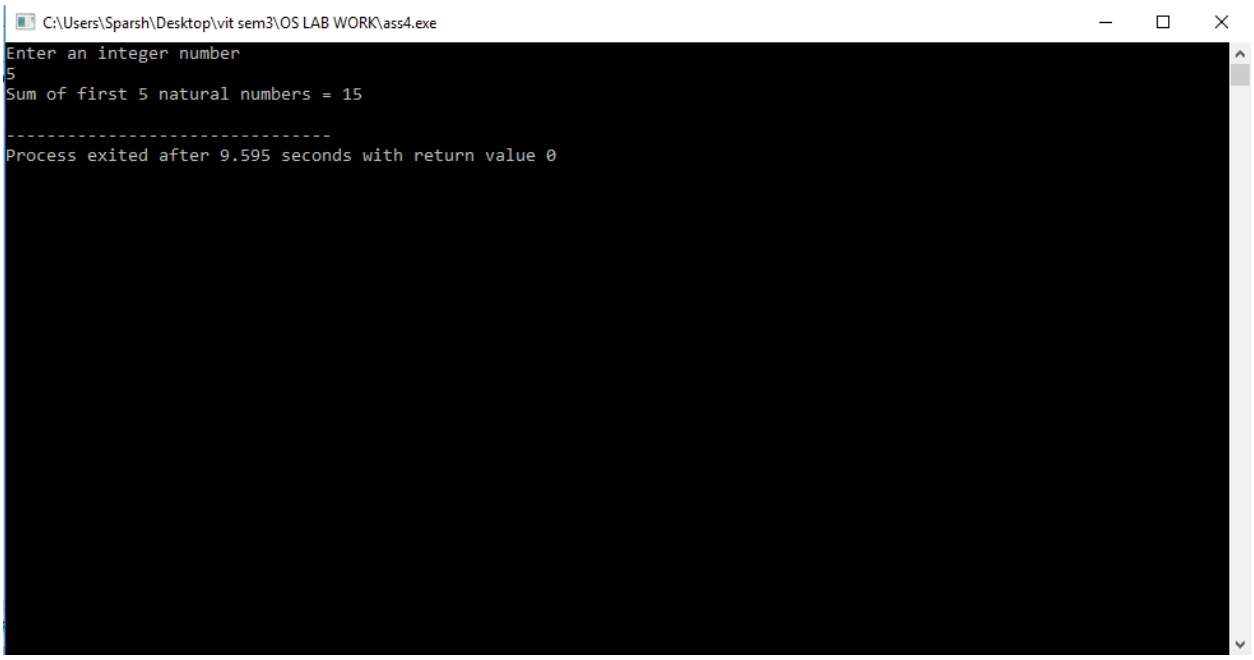
- Take input for a positive number.
- Using a for loop, add all those numbers from 1 to the given number.
- In every cycle, store the value.
- After the loop breaks, display the value.

Source code:

```
#include <stdio.h>


int main()
{
   int i, num, sum = 0;


   printf("Enter an integer number \n");
   scanf ("%d", &num);
   for (i = 1; i <= num; i++)
   {
      sum = sum + i;
   }
   printf ("Sum of first %d natural numbers = %d\n", num, sum);
   return 0;
```

}

Output



Question 9.

Menu driven C program to perform basic arithmetic operations. (Hint: Get corresponding operators and values as input).

Algorithm:

- Take input of a number.

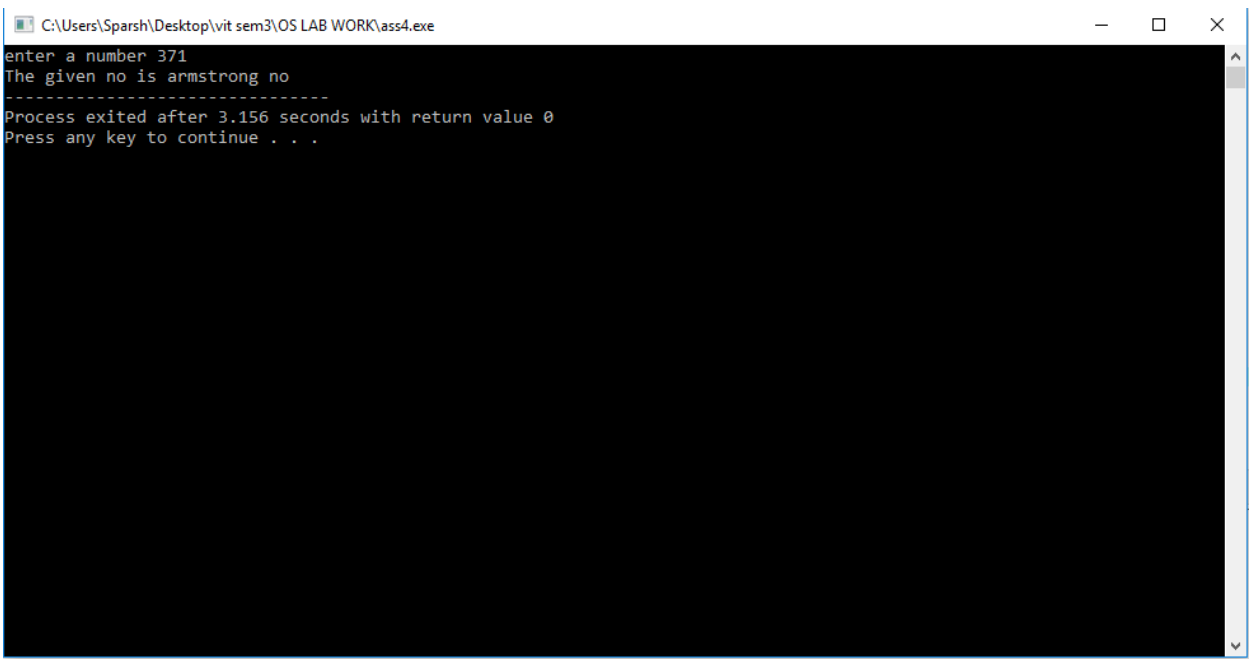
- Break the number into its existing units place, tens place and so on digits.
- Find the cube of all these digits.
- If the sum of the cubes of the digits is equal to the number, then display that the number is an Armstrong number.

Source code:

```
#include <stdio.h>

#include <math.h>


int main()

{

   int number, sum = 0, rem = 0, cube = 0, temp;


   printf ("enter a number");

   scanf("%d", &number);

   temp = number;

   while (number != 0)

   {

      rem = number % 10;

      cube = pow(rem, 3);

      sum = sum + cube;

      number = number / 10;

   }
```

```
    if (sum == temp)
        printf ("The given no is armstrong no");
    else
        printf ("The given no is not a armstrong no");
        return 0;
}
```

Output



Question 10.

C program to check whether given number is perfect square.

Algorithm:

- Take input of a number.
- Run a loop saying from 1 to the number.
- If the product of 2 same numbers between 1 and the given number is equal to the given number, then the given number is a perfect square.
- Else the given number is not a perfect square.
- Display the result.

Source code:

```
#include <stdio.h>


int main()
{
   int a, n;
   printf("Enter a number: ");
   scanf("%d", &n);
   for(a = 0; a <= n; a++)
   {
      if (n == a * a)
      {
         printf("YES");
         return 0;
```

```
        }
    }
    printf("NO");
    return 0;
}
```

Output