

**Exercise 9****Elementary Algorithms**

**9. Write a C++ program for implementation of following elementary algorithms.**

- a) Compute GCD (greatest common divisor) of two given positive integers.**
- b) Factorial of given number using recursion**
- c) Procedure to calculate sum of squares of two integer numbers**
- d) Program to find the maximum and minimum in set of n integer elements.**

**Objective:** the objective of this exercise enable you to perform GCD of two numbers, factorial of given number, sum of squares of two integer numbers, maximum and minimum in set of n integer elements.

**Procedure and Description:****a) GCD of two positive integers:**

In mathematics, the greatest common divisor (gcd), also known as the greatest common factor (gcf), of two non-zero integers, is the largest positive integer that divides the numbers without a remainder. For example, the GCD of 8 and 12 is 4.

**Algorithm GCD – Euclid (m, n)**

//m,n:Positive integers

This algorithm computes the greatest common divisor of two given positive integer

Step 1: begin {of algorithm}

Step 2: while  $n \neq 0$  do

begin {of while loop}

$r \leftarrow m \bmod n$ ; {r: a new variable is used to store the remainder  
which is obtained by dividing m by n, with  $0 \leq r < m$ }

$m \leftarrow n$ ;

{the value of n is assigned as new value of m; but at this  
stage, value of n remains unchanged}

$n \leftarrow r$ ,

{the value of r becomes the new value of n and the value

```
        of r    remains unchanged}
    end {of while loop}
```

Step 3: return (n)

Step 4: end; {of algorithm}

**Expected Output:**

After executing the program enter two positive integer numbers to find gcd.  
Resultant value is output.

Example: Enter two positive integers: 20 16

Output: GCD of two numbers is: 4

**b) Factorial of given number using recursion:**

In a simple language we can define recursion as a programming technique in which a function may call itself.

**Algorithm:** Procedure factorial (n) //n:integer number

Step 1: fac; integer:

```
    begin
```

Step 2: fac ← 1

Step 3: if n equals 1 then return fac

Step 4: else begin

```
    fac ← n * factorial (n – 1)
```

```
    return (fac)
```

```
end;
```

Step 5: Exit

**Expected Output:**

After executing the programs enter an integer number .Program takes the input and executes the factorial recursion function then final factorial function value is resultant output.

Example: enter a number: 4

Output: factorial of given number is: 24

**c) Sum of squares of two integer numbers**

This program gives the steps to calculate sum of squares of two integer numbers

**Algorithm:**

Step 1: Procedure sum-square (a, b: integer) integer;

{denoted the inputs a and b integers and the output is also an integer}

Step 2: S: integer;  
          {to store the required number}  
Step 3: begin  
Step 4:  $S \leftarrow a^2 + b^2$   
Step 5: Returns (S)  
Step 6: end

**Expected Output:**

After executing the program enter two integer numbers then by performing algorithm steps, it calculates the sum of squares of two integer numbers, the resultant value is output.

Example: enter two integer numbers: 6 8

Output: sum of squares of two integer numbers: 100

**d) The maximum and minimum in set of n integer elements**

This program gives the steps for finding maximum value and minimum value in set of n integer elements.

**Algorithm:**

Step 1: procedure SMAXMIN (max, min) //max,min:integer numbers  
Step 2: integer i, max, min; //i: used as index  
Step 3: max  $\leftarrow$  min  $\leftarrow$  A(1); //A: contains set of elements  
Step 4: for I  $\leftarrow$  2 to n do  
Step 5: if A(i)>max then max  $\leftarrow$  A(i)  
Step 6: else if A(i)<min then min  $\leftarrow$  A(i)  
Step 7: end if  
Step 8: end if  
Step 9: repeat  
Step 10: end SMAXMIN

**Expected Output:**

After executing the program. Enter set of n integer numbers. Before that enter n value. Then program executes algorithm steps and gives the output values i.e. the maximum value and minimum value.

Example: Enter n value: 8

Enter set of integer values: 4 8 12 16 3 10 21 7

Output: the maximum and minimum values: 21 3