



LABORATORY WORK BOOK

Name of the Student : Mahek Tabeen
Class.....CSE - C.....Semester.....III
Course Code : AIT.W.O.2.....Course Name : P.W.O. Lab.....
Name of the Course Faculty.....Dr. Mr. D. Suresh Kumar.....Faculty ID : IARE 110.68
Exercise Number : 4.....Week Number : 4.....Date : 8/11/24

| Roll Number | | | | | | |
|-------------|---|---|---|---|---|--------|
| 2 | 3 | 9 | 5 | 1 | A | 0.54 U |

| S. No. | Exercise Number | EXERCISE NAME | MARKS AWARDED | | | | | |
|-----------|--------------------|------------------------------|---------------------|------------------------|----------------------------|-------------|-------------------|----------------|
| | | | Aim/ Preparation | Algorithm / Procedure | | Source Code | Program Execution | Viva - Voce |
| | | | | Performance in the Lab | Calculations and Graphs | | | |
| 1 | 4.1 | Glimming stairs | 4 | 4 | | 4 | 4 | 4 20 |
| 2 | 4.2 | Priore Number Utility | | | | | | |
| 3 | 4.3 | Zodiac sign Calculation | | | | | | |
| 4 | 4.4 | Days in Month calculator | | | | | | |
| 5 | 4.5 | Date validation | | | | | | |
| 6 | 4.6 | Age difference calculator | | | | | | |
| 7 | 4.7 | Elapsed time calculator | | | | | | |
| 8 | 4.8 | Count of specific week days. | | | | | | |
| 9 | 4.9 | Polymorphic printing. | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |

Signature of the Student

Signature of the Faculty

4.1

Climbing Stairs.

using System;

Public class Program

{

Public static void main()

{

Console.WriteLine(ClimbStairs(2));

Console.WriteLine(ClimbStairs(3));

Console.WriteLine(ClimbStairs(5));

Console.ReadKey();

}

Public static int ClimbStairs(int n)

{

if (n == 1)

return 1;

if (n == 2)

return 2;

int[] dp = new int[n + 1];

dp[1] = 1;

dp[2] = 2;

for (int i = 3; i <= n; i++)

{

dp[i] = dp[i - 1] + dp[i - 2];

}

return dp[n];

}

}

Actual Output:

2

3

8

4.2

Prime Number Utility

using System;

Public class PrimeProgram

{

Public static void main()

{

Console.WriteLine("Select an option:");

Console.WriteLine("1. Check if a number is prime");

 Console.WriteLine("2. Find the nth prime number"); Console.WriteLine("3. Display prime numbers up to
 a given limit");

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Enter the number to check:");

int numberToCheck = int.Parse(Console.ReadLine());

if (!IsPrime(numberToCheck))

{

Console.WriteLine(\$"{numberToCheck} is prime.");

} else {

Console.WriteLine(\$"{numberToCheck} is Not Prime.");

}

break;

Case 2:

Console.WriteLine("Enter the Value of n: ");

int n = int.Parse(Console.ReadLine());

int nthPrime = FindNthPrime(n);

 Console.WriteLine(\$"The {n}th prime number is
 {nthPrime}.");

break;

Case 3:

Console.WriteLine("Enter the limit:");

int limit = int.Parse(Console.ReadLine());

Console.WriteLine(\$"Prime numbers up to {limit}:");

PrintPrimesSeries(limit);

break;

default:

Console.WriteLine("Invalid choice.");

break;

}

Console.ReadKey();

Public static bool IsPrime(int number)

{

if (number <= 1) return false;

if (number <= 3) return true;

if (number % 2 == 0) || (number % 3 == 0) return false;

for (int i = 5; i * i <= number; i += 6)

{

if (number % i == 0 || number % (i + 2) == 0)

{

return false;

}

}

return true;

}

Public static void int FindNthPrime(int n)

{

if (n < 1) throw new ArgumentException("n must be a +ve integer");

int count = 0;

int number = 2;

while (true)

```

if (IsPrime (number)) {
    count++;
    if (count == n)
    {
        return number;
    }
    number++;
}

```

```

Public static void PrintPrimeSeries (int limit)
{
    if (limit < 2)
    {
        Console.WriteLine ("No primes in this range .");
        return;
    }
    for (int i = 2; i <= limit; i++)
    {
        if (IsPrime (i))
        {
            Console.Write (i + " ");
        }
    }
    Console.WriteLine ();
}

```

Actual Output:

Task 1: select an option:

1. check if a. number is prime
2. Find the nth prime number
3. Display prime numbers up to a given limit

Enter your choice (1/2/3) : 1


```

case 8: return (day <= 22) ? "Leo" : "Virgo";
case 9: return (day <= 22) ? "Libra" : "Scorpio";
case 10: return (day <= 21) ? "Scorpio" : "Sagittarius";
case 11: return (day <= 21) ? "Sagittarius" : "Capricorn";
default: throw new ArgumentOutOfRangeException("Month is out of Range");
}

```

```

public static bool TryParseDate(string input, out DateTime birthDate)
{
    return DateTime.TryParseExact(input, "dd/MM/yyyy", null,
        System.Globalization;
}

```

Output:

Enter your date of birth (DD/MM/YYYY):
01/01/2024

Your Zodiac sign is: Capricorn.

4.4

Days in a Month Calculator

using System;

public class DayInMonthCalculator

{

public static void Main()

{

Console.WriteLine("Enter the month (1 for January & for February, etc):");

if (int.TryParse(Console.ReadLine(), out int month))

& & month >= 1 & & month <= 12

{

Console.WriteLine("Enter the year:");

if (int.TryParse(Console.ReadLine(), out int year) & & year > 0)

{

```

        int days = GetDaysInMonth (Month, year);
        Console.WriteLine ("The number of days in
                           {GetMonthName (Month)} {year} is {days}.")
    }

    else {
        Console.WriteLine ("Invalid year: please enter a
                           positive integer")
    }

    }

    else {
        Console.WriteLine ("Invalid Month: Plz enter a
                           number b/w 1 & 12");
    }

    }

    Console.ReadKey ();
}

```

```

public static bool IsLeapYear (int month, int year)
{
    return (year % 4 == 0 & & year % 100 != 0) || (year % 400
                                                 == 0);
}

public static string GetMonthName (int Month)
{
    string [] months = {"January", "February", "March",
                        "April", "May", "June", "July", "August",
                        "September", "October", "November", "December"};
    return month[Month - 1];
}

```

Output:

Enter the month (1 for January, 2 for February, etc):

2
Enter the Year:

2024

The number of day in February 2024 is 29.

4.5

Date Validator

using System;

Public class DateValidator

{ Public static void Main()

Console.WriteLine("Enter a date in the format DD/MM/YYYY");
string input = Console.ReadLine();

If (IsValidDate(input, out int day, out int month, out int year))

Console.WriteLine("The data is Valid");

y

else {

Console.ReadKey();

y

Public static bool IsValidDate(string date, out int day,
out int month, out int year)

d

day = 0;

month = 0;

year = 0;

string[] parts = data.Split('/');

if (parts.Length != 3) {

return False

y

If (!int.TryParse(parts[0], out day) || !int.TryParse

(parts[1], out month) || !int.TryParse(parts[2], out year))

d

return False;

y

If (year < 1) { return False; } y

If (month < 1 || month > 12) {

return False;

y

```

int daysInMonth = GetDaysInMonth(month, year);
if (day < 1 || day > dayInMonth) {
    return False;
}
return True;

```

```
Public static int GetDaysInMonth(Month, year);
```

```

if (month == 2) {
    return IsLeapYear() ? 29 : 28;
}
if (month == 4 || month == 6 || month == 9 || month == 11)
{
    return 30;
}
return 31;

```

```
Public static bool IsLeapYear(int year);
```

```

return (year % 4 == 0 && year % 100 != 0) ||
       (year % 400 == 0);

```

Output:

Enter a date in the format DD/MM/YYYY;
30/02/2024

The date is invalid.

4.6

Age Difference Calculator

using System;

Public class AgeDifferenceCalculator

```
public static void Main()
```

```

Console.WriteLine("Enter the first birth date (DD/MM/YYYY)");
string FirstDateInput = Console.ReadLine();
Console.WriteLine("Enter the second birth date");

```

```

String SecondDateInput = Console.ReadLine()
(DataTime.TryParseExact(FirstDateInput, "dd/MM/yyyy"))
    null, System.Globalization.DateTimeStyles.None,
    out DateTimeSecondDate))

{ int year, int month, int days) = CalculateAgeDiff
    (firstDate, secondDate);
Console.WriteLine("Age Difference : {years} years,
    {months} months, {days} days");
}

else {
    Console.WriteLine("Invalid date format plz use
        DD/MM/yyyy")
}

}

Public static (int) CalculateAgeDifference (DateTime,
    date1, DateTime, date2)
{
    if (date1 > date2)
    {
        DateTime temp = date1;
        date1 = date2;
        date2 = temp;
    }

    int years = date2.Year - date1.Year;
    int months = date2.Month - date1.Month;
    int days = date2.Day - date1.Day;
    if (days < 0)
    {
        months--;
        days += DateTime.DaysInMonth(date1.Year, date1.Month);
    }

    if (months < 0)
    {
}

```

```

year --;
month += 12;
}
return (years, months, days);
}

```

Output:

01/01/2024

01/01/2023

Age difference : 1 year, 0 months, 0 days.

4.7 Elapsed Time Calculator.

using System;

public class ElapsedTimeCalculator

{ public static void Main()

{

```

Console.WriteLine("Enter the first time stamp (HH MM ss)");
string[] firstTimeInput = Console.ReadLine().Split();
Console.WriteLine("Enter the second time stamp (HH MM ss):");
string[] secondTimeInput = Console.ReadLine().Split();
if (TryParseTime(firstTimeInput, out SpanFirst)
    && TryParseTime(secondTimeInput, out SpanSecond))
    {
        int hours, int minutes, int seconds = Calculate.Elapsed
            (firstTime, secondTime);
        Console.WriteLine($"Elapsed Time : {hours} hours, {minutes} minutes, {seconds} seconds");
    }
    else
        Console.WriteLine("Invalid time format plz use HH MM ss with valid ranges");
    }
    
```

```
Private static void TryParseTime (String[], timeInput,
                                out TimeSpan time)
{
```

```
    if (timeInput.length == 3 & &
        int.TryParse(timeInput[0], out int hours) & &
        int.TryParse(timeInput[1], out int minutes) & &
        hours >= 0 & & hours < 24 & &
        minutes >= 0 & & minutes < 60 & &
        seconds >= 0 & & seconds < 60
    {

```

```
        time = new TimeSpan(hours, minutes, seconds)
        return true;
    }
```

```
    time = TimeSpan.zero;
    return false;
}
```

```
Private static (int, int, int) calculate Elapsed Time
    (TimeSpan startTime, TimeSpan endTime)
{
```

```
    if (endTime < startTime)
```

```
        endTime = endTime.Add(new TimeSpan(4, 0, 0) next day)
```

```
y.
```

```
TimeSpan elapsed = endTime - startTime;
```

```
return (elapsed, hours, elapsed.Minutes, elapsed.Seconds);
```

```
y
```

Output:

Enter the first timestamp (HH MM SS);

10 10 80

Enter the second timestamp (HH MM SS);

10 10 80

Elapsed time : 0 hours, 0 minutes, 0 seconds;

4.8 Count Of Specific Weekdays

using System;
 using System.Globalization;
 public class CountSpecificWeekdays

```

public static void Main()
{
    Console.WriteLine("Enter the first date (DD/MM/YYYY);");
    string startDateInput = Console.ReadLine();
    Console.WriteLine("Enter the second date (DD/MM/YYYY);");
    string endDateInput = Console.ReadLine();
    Console.WriteLine("Enter the week day to count:");
    string weekdayInput = Console.ReadLine();
    if (!TryParseDate(startDateInput, out DateTime startDateTime, endDateTime))
        && Enum.TryParse(weekdayInput, true, out Day)
    {
        int weekdayCount = countWeekDays(startDateTime, endDateTime, Day);
        Console.WriteLine($"The number of {Day} is {weekdayCount}");
    }
    else
    {
        Console.WriteLine("Invalid input plz ensure the date are in DD/MM/YYYY format and the weekday is correctly spelled");
    }
    Console.ReadKey();
}

while (currentDate <= endDate)
{
    if (currentDate.DayOfWeek == Day)
    {
        count++;
    }
}
  
```

```

        currentDate = currentDate.AddDays(1);
}
return count;
}

```

Output:

01/08/2024
03/08/2024
Saturday.

4.9 Polymorphic Pointing.

using System;

public class FunctionOverloadingDemo

{

public void print(int value)

{

Console.WriteLine("Integer Value : " + value);

}

public void print(double value)

{

Console.WriteLine("Floating - point Value : " + value);

}

public void print(string value)

{

Console.WriteLine("String Value : " + value);

}

public static void Main()

{

Console.WriteLine("String Value : " + value);

}

public static void Main()

demo.print(42);

function

demo.print("Hello; world!");
function
Console.ReadKey();

y

y

Output:

42

3.14159

Hello World!

