IARE

IARE
INSTITUTE OF AERONAUTICAL ENGINEERING
(An Autonomous Institute affiliated to JNTUH, Hyderabad),
Dundigal, Hyderabad - 500 043

LABORATORY WORK BOOK

Name of the Student : N. Ravi Chandrika

Class : CSD-B Semester : III rd Semester

Course Code : AITD02 Course Name : PD Laboratory

Name of the Course Faculty : Ms. R. Tejaswini Faculty ID : IARE10999

Exercise Number : 5 Week Number : 5 Date : 11/11/24

Roll Number									
2	3	9	5	1	A	6	7	B	3

S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED						
			Aim/ Preparation	Algorithm / Procedure	Performance in the Lab	Source Code Calculations and Graphs	Program Execution Results and Error Analysis	Viva - Voce	Total
			4	4		4	4	4	20
1	5.1	Is the phone no-formatted correctly	4	2	2	4	4	4	20
2	5-2	Basic E-mail validation	4	2	2	4	4	4	20
3	5-3	Valid C# comments	4	2	2	4	4	4	20
4	5-4	IPv4 validation	4	2	2	4	4	4	20
5	5-5	Password validation	4	2	2	4	4	4	20
6	5-6	Valid ttexcode	4	2	2	4	4	4	20
7	5-7	Retrieve hostname	4	2	2	4	4	4	20
8	5-8	Retrive hostname & IPAddress	4	2	2	4	4	4	20
9	5-9	Retrive IPAddress of a Domain name	4	2	2	4	4	3	19
10	5-10	Retrive IPv4, IPv6 Address & check address family	4	2	2	4	4	3	19
11	5-11	Retrive hostname based on IPAddress.	4	2	2	4	4	4	20
12									

N. Ravi Chandrika
Signature of the Student

Signature of the Faculty

ROLL NUMBER:

START WRITING FROM HERE

5-1 Aim: Create a function that accept a string and returns true if it's in the format of a proper phone number and false if it's not.

Code:

```
using System;
using System.Text.RegularExpressions;
public class Program
{
  public static bool IsValidPhoneNumber(string phoneNumber)
  {
    string pattern = @"^\(\d{3}\) \d{3}-\d{4}$";
    return Regex.IsMatch(phoneNumber, pattern);
  }
  public static void Main()
  {
  Console.WriteLine(IsValidPhoneNumber("(123) 456-7890"));
  Console.WriteLine(IsValidPhoneNumber("(111) 555-2345"));
  Console.WriteLine(IsValidPhoneNumber("(098) 123-4567"));
  }
}
```

Output:

True

False

False.

2 Aim: Create a function that accepts a string, checks if it's a valid email address and returns either true of false, depending on the evaluation.

ROLL NUMBER :

Code:

```
using System;
Public Class Program
{
  public static bool ValidateEmail(string email)
  {
    int atIndex = email.IndexOf('@');
    if(atIndex <=0)
    return false;
    int dotIndex = email.IndexOf('.', atIndex +1);
    if (dotIndex <= atIndex+1)
      return false;
    if (dotIndex == email.length -1)
      return false;
  }
  return true;
}
public static void Main()
{
  Consde.WriteLine(ValidateEmail("@gmail.com"));
  Console.WriteLine(ValidateEmail("hello@gmail.com"));
  Console.WriteLine(ValidateEmail("gmail"));
  Console.WriteLine(ValidateEmail("hello@gmail"));
  Console.WriteLine(ValidateEmail("hello@dabit.com"));
  }
}
```

Output: False

False False

False True.

5.3 Aim: In C#, consides two types of comments:

- Single line comments Staits with "//"
- Multi line or block comments Staut with /* & end with */.

Code:

```
using System;
public static bool CommentsCorrect(String input)
{
    int i=0;
    while (i<input.Length)
    {
      if(i+1<input.Length && input[i]=='/' &&
                                  input[i+1]=='/')
      {
          i+=2;
      }
      else if(i+1<input.Length && input[i]=='/' &&
                                  input[i+1]=='*')
      {
          i+=2;
      }
      if(i+1<input.Length && input[i]=='*' &&
                                  input[i+1]=='/')
      {
          i+=2;
      }
```

```
    else
    {
        return false;
    }
}
else
{
    return false;
}
}
return true;
}
Console.WriteLine(CommentsCorrect("//////"));
Console.WriteLine(CommentsCorrect("/**//**////**/"));
Console.WriteLine(CommentsCorrect("///*/**/"));
Console.WriteLine(CommentsCorrect("////"));
```

Output:

True

True

False

False.

5.4 Aim: Create a function that takes a string and returns true if the IP is valid or false if its not.

a string representing the IPv4 Address.

Code:

```
Using System;
Public class IPAddressValidator {
  Public Static bool IsValidIP(string ip) {
    String[] parts = ip.split('.');
    if (parts.Length != 4)
      return false;
    for (int i=0; i< parts.Length; i++) {
      String part = parts[i];
      if (!int.TryParse (part, out int num))
        return false;
      if (num < 0 || num > 255)
        return false;
      if (part.length > 1 && part[0] == '0')
        return false;
      if (i == 3 && num == 0)
        return false;
    }
    return true;
}
public Static void Main()
{
    Consde.WriteLine (IsValidIP("1.2.3.4"));
    Consde.WriteLine (IsValidIP("1.2.3"));
    Console.WriteLine (IsValidIP("1.2.3.4.5"));
    Console.Writeline (IsValidIP("123.45.67.89"));
```

```
        Console.WriteLine(IsValidIP("123.456.78.90"));
        Console.WriteLine(IsValidIP("123.045.067.089"));
    }
}
```

Output:

True	True
False	False
False	False

5.5 Ans: Create a function that validates a password based on the following rules, given.

Code:

```
using System;
using System.Linq;
using System.Text.RegularExpressions;
public class PasswordValidator
{
public static bool ValidatePassword(string password)
  { if (password.Length < 6 || password.Length > 24)
      return false;
    if (!password.Any(char.IsUpper))
      return false;
    if (!password.Any(char.IsLower))
      return false;
    if (!password.Any(char.IsDigit))
      return false;
```

```
if(Regex.IsMatch(password, @"[0-9]{2,}"))
    return false;
String validSpecialChars = "!@#$%^&*()+=-{}[]:;|'?<>";
foreach (char c in password)
{
    if(!char.IsLetterOrDigit(c) && !validSpecialChars.Contains(c))
        return false;
}
return true;
}
public static void Main()
{
Console.WriteLine(ValidatePassword("P1zz@"));
Console.WriteLine(ValidatePassword("iloveyou"));
Console.WriteLine(ValidatePassword("Fhg93@"));
}
}
```

Output: False
False
True.

5.6 Aim:

Create a function that determines, whether a string is valid hexcode. A hexcode must begin with a pound key # and is exactly 6 characters is length. Each character must be a digit from 0-9 or an alphabet character from A-F. All alphabetic characters may be uppercase or lowercase.

ROLL NUMBER :

Code:

```
using System;
using System.Text.RegularExpressions;
public class ItemCodeValidator
{
    public static bool IsValidItemCode(string code)
    {
        string pattern = @"^[A-Za-z]{3}-\d{3}$";
        return Regex.IsMatch(code, pattern);
    }
    public static void Main()
    {
        Console.WriteLine("IsValidItemCode("ABC-123"));
        Console.WriteLine("IsValidItemCode("abc-123"));
        Console.WriteLine("IsValidItemCode("ABC-1234"));
        Console.WriteLine("IsValidItemCode("AB-123"));
        Console.WriteLine("IsValidItemCode("ABC-12a"));
    }
}
```

Output:

True
True False
True False.

5-7 Aim:

Create a function that retrieves the hostname of the local machine. The function should return the hostname of the machine on which code is running.

- No input is required for this function.

Doubly Linked List (Forward Traversal):

ROLL NUMBER :

Code:

```
using System;
using System.Net;
public class HostNameRetrieval
{
  public static string GetLocalHostName()
  {
    return DNS.GetHostName();
  }
  public static void Main()
  {
    Console.WriteLine(GetLocalHostName());
  }
}
```

Output: DELL2055-

8. Aim:

Create a function that retrieves hostname and the IP Addresses of the local machine. The function should display the hostname and all associated IP Addresses.

Code:

```
using System;
using System.Net;
public class NetworkInfo
{
  public static void DisplayHostNameAndIPAddresses()
  {
    string hostName = Dns.GetHostName();
    Console.WriteLine("Host Name of machine = " + hostName);
```

ROLL NUMBER:

```
IPAddress[] addresses = Dns.GetHostAddresses(hostName);
Console.WriteLine("IP Addresses of Machine:");
foreach (IPAddress address in addresses)
{
    Console.WriteLine(address.ToString());
}
}
public static void Main()
{
    DisplayHostNameAndIPAddresses();
}
}
```

Output:

Host Name of Machine = DELL2055

IP Addresses of Machine : fe80::2010:b572:61dh:c150%4,

192.168.1.10

5-] Aim:

Create a function that retrieves the IP addresses of the given domain name.

Code:

```
using System;
using System.Net;
public class DomainIPRetriever
{
  public static void GetIPAddresses(String domainName)
  {
    try
    {
      IPAddress[] addresses = Dns.GetHostAddresses();
```

Output:

ROLL NUMBER :

```
        Console.WriteLine(Addresses) of {domainName}:");
        foreach(IPAddress address in addresses)
        {
            Console.WriteLine(address.ToString());
        }
    }
    catch(Exception ex)
    {
        Console.WriteLine($"Error retrieving IP Addresses");
    }
}
public static void Main()
{
    GetIPAddresses("www.bing.com");
}
}
```

Output:

```
IP Address(es) of www.bing.com:
    23.65.124.99
    23.65.124.97
```

5.10 Aim:

Create a function that retrieves and displays both IPV4 and IPV6 addresses of local machine separately by checking the address family. The function should first output the host name of the machine, followed by a separate lists of the IPV4 & IPV6 addresses.

Code:

```
using System;
using System.Net;
using System.Linq;
public class localIPRetrieves
{ public static void DisplayHostNameAndIPAddresses()
  {
    string hostName = Dns.GetHostName();
    Console.WriteLine("Host Name of machine :" + hostName);
    IPAddress[] addresses = Dns.GetHostAddresses(hostName);
    var ipv4Addresses = addresses.Where(ip => ip.AddressFamily);
    Console.WriteLine("IPv4 Addresses of Machine:");
    foreach(IPAddress ipv4 in ipv4Addresses)
    {
      Console.WriteLine(ipv4.ToString());
    }
    var ipv6Addresses = addresses.Where(ip => ip.AddressFamily);
    Console.WriteLine("IPv6 Addresses of Machine:");
    foreach(IPAddress ipv6 in ipv6Addresses)
    {
      Console.WriteLine(ipv6.ToString());
    }
  }
  public static void Main()
  {
    DisplayHostNameAndIPAddresses();
  }
}
```

Output:

ROLL NUMBER :

Output:

Host Name of Machine = Dell3055

IPv4 Address(es) of Machine : 192.168.1.10

IPv6 Address(es) of Machine : fe80::3010:7b78:61db:c75d%14

5.11 Aim:

Create a function that takes an IPAddress as input and returns the hostname associated with that IP Address.

Code:

```
using System;
using System.Net;
public class IPToHostName
{
    public static void DisplayHostName(string ipAddress)
    {
        try
        {
            IPHostEntry hostEntry = Dns.GetHostEntry();
            Console.WriteLine("Host Name of IP Address " +
                ipAddress + " is: " + hostEntry.HostName);
        }
        catch(Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
    public static void Main()
```

```
        {
            DisplayHostName("204.79.197.200");
        }
    }
```

Output:

Host Name of IPAddress 204.79.197.200 is :

a-0001.a-msedge.net.