INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
| 01 | 14-10-21 | 01 | Arrays |  |
| 02 | 21-10-21 | 02 | Linear Search & Sorting |  |
| 03 | 28-10-21 | 03 | Linked List |  |
| 04 | 4-11-21 | 04 | Doubly & Circular Linked List |  |
| 05 | 11-11-21 | 05 | Stacks |  |
| 06 | 18-11-21 | 06 | Recursion |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**05**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | **Write a program to build your own stack class. The minimum your stack class should include is using your enrollment no :**   * + - **A Push(Object) method**     - **A Pop() method**     - **A Peek() method**     - **A Is Full () method**     - **A Is Empty () method**     - **A Display() method**     - **A Count() method** |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

2-Dec-21

**Task No. 1: Write a program to build your own stack class. The minimum your stack class should include is using your enrollment no:**

* + - **A Push (Object) method**
    - **A Pop () method**
    - **A Peek () method**
    - **A Is Full () method**
    - **A Is Empty () method**
    - **A Display () method**
    - **A Count () method**

**Solution:**

internal class Stack

{

static readonly int Max = 1000;

int top;

int[] stack = new int[Max];

bool IsEmpty()

{

return (top < 0);

}

public Stack()

{

top = -1;

}

internal bool Push(int data)

{

if (top >= Max)

{

Console.WriteLine("Stack OverFlow");

return false;

}

else

{

stack[++top] = data;

return true;

}

}

internal int pop()

{

if (top < 0)

{

Console.WriteLine("Stack UnderFlow");

return 0;

}

else

{

int value = stack[--top];

return value;

}

}

internal void peak()

{

if (top < 0)

{

Console.WriteLine("Stack UnderFlow");

return;

}

else

{

int value = stack[top];

Console.WriteLine("The Top Most Element Of Stack Is : {0} ", stack[top]);

}

}

internal void PrintStack()

{

if (top < 0)

{

Console.WriteLine("Stack UnderFlow");

return;

}

else

{

int count = 0;

Console.WriteLine("Items In Stack Are ");

for (int i = top; i >= 0; i--)

{

Console.WriteLine(stack[i]);

count++;

}

Console.WriteLine("Count : " + count);

}

}

static void Main(string[] args)

{

Stack mystack = new Stack();

mystack.Push(10);

mystack.Push(20);

mystack.Push(30);

mystack.Push(40);

mystack.PrintStack();

mystack.pop();

Console.WriteLine("");

Console.WriteLine("AFTER POP ");

Console.WriteLine("");

mystack.PrintStack();

Console.WriteLine("");

mystack.peak();

Console.WriteLine("\n");

mystack.PrintStack();

Console.ReadKey();

**OUTPUT**:

Text

Description automatically generated

Bahria University,

Karachi Campus

A picture containing text, room

Description automatically generated

LAB EXPERIMENT NO.

**06**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | **Write a code which prints the following series:**  **2 4 8 - - - - n** |
| **02** | **Write a program which takes input of an integer number and returns the sum of all numbers. i.e., if input is 3453 then the output should be 15 (3+4+5+3).** |
| **03** | **Write a program to calculate binomial coefficients of any given number using recursion.** |
| **04** | **Calculation of number of moves for N number of disks in Tower of Hanoi problem using recursion.** |
| **05** | **Write a program to calculate H.C.F of two numbers, using recursion.** |
|  |  |
|  |  |

Submitted On:

2-Dec-21

**Task No. 1: Write a code which prints the following series:**

**2 4 8 - - - - n.**

**Solution:**

public static void series(int a) //5,4,3,2,1

{

if (a > 1)

{

//5-1=4,3,2,1

series(a - 1);

Console.Write("{0} ", Math.Pow(2, (a - 1)));

}

}

static void Main(string[] args)

{

series(8);

Console.WriteLine();

Console.ReadLine();

}

**OUTPUT**:

Text

Description automatically generated

**Task No. 2: Write a program which takes input of an integer number and returns the sum of all numbers. i.e., if input is 3453 then the output should be 15 (3+4+5+3).**

**Solution:**

int[] a = new int[5];

int sum = 0;

Console.WriteLine("Enter Numbers");

for (int i = 0; i < a.Length; i++)

{

Console.Write("Enter Number At {0} : " , i);

a[i] = Convert.ToInt32(Console.ReadLine());

sum += a[i];

}

for (int i = 0; i < a.Length; i++)

{

Console.Write("\nSum : {0}({1}+{2}+{3}+{4}+{5}) ", sum, a[0], a[1], a[2], a[3], a[4]);

break;

}

Console.WriteLine();

Console.ReadLine();

**OUTPUT**:

Text

Description automatically generated

**Task No. 3: Write a program to calculate binomial coefficients of any given number using recursion.**

**Solution:**

public static long nfact(long a)

{

if (a == 0)

{

return 1;

}

return a \* nfact(a - 1);

}

static void Main(string[] args)

{

long n, k, t;

long binomial;

Console.WriteLine("\nN Must Be Greater Than K ");

Console.Write("\n\nEnter Value Of N : " );

n = Convert.ToInt64(Console.ReadLine());

Console.Write("Enter Value Of K : ");

k = Convert.ToInt64(Console.ReadLine());

binomial = nfact(n) / (nfact(k) \* nfact(n - k));

Console.WriteLine("\nBinomial Co-efficient : {0} " , binomial);

Console.ReadLine();

}

**OUTPUT**:

Text

Description automatically generated

**Task No. 4: Calculation of number of moves for N number of disks in Tower of Hanoi problem using recursion.**

**Solution:**

public static int moves(int disc)

{

if (disc == 0)

{

Console.WriteLine("Enter Above Value ");

return 0;

}

else if (disc == 1)

{

return 1;

}

else

{

return 2 \* moves(disc - 1) + 1;

}

}

static void Main(string[] args)

{

Console.Write("Enter No Of Discs : ");

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

Console.WriteLine("No Of Moves Are : " + moves(discs));

Console.WriteLine();

Console.ReadLine();

}

**OUTPUT**:

Text

Description automatically generated

**Task No. 5: Write a program to calculate H.C.F of two numbers, using recursion.**

**Solution:**

static int hcf(int a, int b)

{

// Everything divides 0

if (a == 0)

return b;

if (b == 0)

return a;

// base case

if (a == b)

return a;

// a is greater

if (a > b)

return hcf(a - b, b);

return hcf(a, b - a);

}

static void Main(string[] args)

{

int a, b;

Console.Write("Enter Value Of A : ");

a = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter Value Of B : ");

b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("\nHCF Of " + a + " & " + b + " Is : " + hcf(a, b));

Console.WriteLine();

Console.ReadLine();

}

**OUTPUT**:

Text

Description automatically generated