 **BAHRIA UNIVERSITY (KARACHI CAMPUS)**

**(OEL-I) - Fall 2021**

# DATA STRUCTURES

Class: **BSE 3(B)**

Course Instructor:  **Ma’am Laraib Submission Deadline: 25th Nov, 2021**

Lab Instructor: **Engr. Ayesha Khan** Max Marks: 30 CLO mapping: CLO 1

Name : Muhammad Junaid Saleem Qadri E.no: 02-131202-057

Note: submit your file in proper format starting from your name ending with enrollment last digits [Abc001]

**Question 1: [CLO1,30.0 Marks]**

Q1: Write a program in C# to create a doubly linked list and display in reverse order

Q2: Create a stack of ages of student by taking user input. Once the stack is full ,sort them by RECURSIVE method.

Q3: create an array of 5 numbers(by using user input). If you find square of first value in an array , convert it into cube of that number. Now create new array2 of length 5,starting from the value of that cube followed by increment by 2 for the others value.

**Note**: If you submit your assignment after the given timings then 5 **Marks** will be deducted for the late submissions (i.e. 25th Nov, 2021 ) and after 25th Nov, 2021 , **Zero** will be awarded for those who would not be able to submit till then.

**Question :1**

**Solution :**

**Class Linked\_list**

class Linked\_List

{

public Node head;

public void printlist(Node node)

{

Node last = null;

Console.WriteLine("\nTraversal in forward Direction");

while (node != null)

{

Console.Write(node.data + " ");

last = node;

node = node.next;

}

Console.WriteLine();

Console.WriteLine("Traversal in reverse direction");

while (last != null)

{

Console.Write(last.data + " ");

last = last.prev;

}

}

}

**Class Node**

class Node

{

public Node next;

public int data;

public Node prev;

public Node(int d)

{

next = null;

prev = null;

data = d;

}

}

**Main method**

static void Main(string[] args)

{

Linked\_List list = new Linked\_List();

Console.Write("Enter the First number : ");

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

Console.Write("Enter the Second number : ");

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

Console.Write("Enter the Third number : ");

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

list.head = new Node(one);

Node Second = new Node(second);

Node Third = new Node(third);

list.head.next = Second;

Second.prev = list.head;

Second.next = Third;

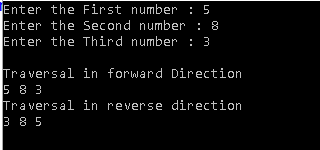
Third.prev = Second;

list.printlist(list.head);

Console.ReadLine();

}

**Output:**



**Question :2**

**Solution :**

public static void bubbleSort(int[] arr, int n)

{

if (n == 1)

return;

for (int i = 0; i < n - 1; i++)

if (arr[i] > arr[i + 1])

{

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

bubbleSort(arr, n - 1);

}

static void Main(string[] args)

{

Stack age = new Stack();

for (int i = 1; i <= Stack.max; i++)

{

Console.Write("Age of Student {0} : ", i);

age.userPush(int.Parse(Console.ReadLine()));

}

Console.WriteLine("\n\tBefore Sorting : ");

age.printStack(age.stack);

Console.WriteLine("\n\tAfter Sorting");

bubbleSort(age.stack, 5);

age.printStack(age.stack);

Console.ReadLine();

}

**Stack Class**

public class Stack

{

int top;

public static int max = 5;

public int[] stack = new int[max];

public Stack()

{

top = -1;

}

public bool userPush(int data){

if(top>=max){

Console.WriteLine("Stack Overflow");

return false;

}

else

{

stack[++top] = data;

return true;

}

}

public void printStack(int[] stack)

{

if (top < 0) {

Console.WriteLine("Stack under Flow");

return;

}

else

{

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

{

Console.Write(stack[i]+" ");

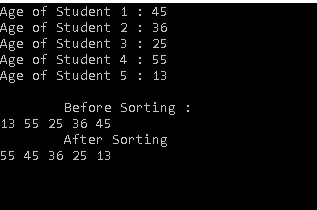
}

}

}

}

**Output :**



**Question :3**

static void Main(string[] args)

{

int[] array = new int[5];

for (int i = 0; i < 5; i++)

{

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

array[i] = int.Parse(Console.ReadLine());

}

for (int i = 0; i < 5; i++)

{

if (i != 4)

{

if (array[i] == (array[i + 1] \* array[i + 1]))

{

array[i] = (array[i + 1] \* array[i + 1] \* array[i + 1]);

}

}

}

Console.WriteLine("after cube");

for (int i = 0; i < 5; i++)

{

Console.Write(array[i]+" ");

}

Console.ReadLine();

}

Output: