**ЗВІТ**

**Основи програмування**

**Лабораторна робота №7**

**ПОБУДОВА ТА ВИКОРИСТАННЯ СТРУКТУР ДАНИХ**

***Виконала: Григор’єва В.І.***

**Київ 2025**

**Завдання:**

A white page with black text

AI-generated content may be incorrect.

**Код програми мовою C#:**

1. **DoublyLinkedListDouble.cs**
2. using System.Collections;  
     
   namespace DoubleLinkedList  
    {  
    public class DoublyLinkedListDouble : IEnumerable  
    {  
    private Node tail = null;  
    private Node head = null;  
     
    public DoublyLinkedListDouble()  
    {  
    tail = null;  
    head = null;  
    }  
     
    public bool IsEmpty()  
    {  
    return head == null;  
    }  
     
    public double? FindMaxValue()  
    {  
    if (IsEmpty()) return null;  
     
    Node current = head;  
    double max = current.Data;  
     
    while (current != null)  
    {  
    if (current.Data > max)  
    {  
    max = current.Data;  
    }  
    current = current.Next;  
    }  
     
    return max;  
    }  
     
    public double? FindMeanValue()  
    {  
    if (IsEmpty())  
    {  
    return null;  
    }  
     
    Node current = head;  
    int count = 0;  
    double sum = 0.0;  
     
    while (current != null)  
    {  
    sum += current.Data;  
    count++;  
    current = current.Next;  
    }  
     
    if (count == 0)  
    {  
    return null;  
    }  
    return sum / count;  
    }  
     
    public void AddElementToStart(double value)  
    {  
    var newElement = new Node(value);  
     
    if (IsEmpty())  
    {  
    head = tail = newElement;  
    }  
    else  
    {  
    newElement.Next = head;  
    head.Prev = newElement;  
    head = newElement;  
    }  
    }  
     
    public double? FindFirstPositiveElement()  
    {  
    var current = head;  
    while (current != null)  
    {  
    if (current.Data > 0.0)  
    {  
    return current.Data;  
    }  
    current = current.Next;  
    }  
     
    return null;  
    }  
     
    public int FindNumberElementsBiggerMeanValue()  
    {  
    double? meanValue = FindMeanValue();  
    if (meanValue == null)  
    {  
    return 0;  
    }  
     
    int count = 0;  
    var current = head;  
     
    while (current != null)  
    {  
    if (current.Data > meanValue)  
    {  
    count++;  
    }  
    current = current.Next;  
    }  
     
    return count;  
    }  
     
    public List<double> GetElementsBiggerMeanValue()  
    {  
    var result = new List<double>();  
    double? meanValue = FindMeanValue();  
      
    if (meanValue == null)  
    {  
    return result;  
    }  
     
    var current = head;  
    while (current != null)  
    {  
    if (current.Data > meanValue)  
    {  
    result.Add(current.Data);  
    }  
    current = current.Next;  
    }  
     
    return result;  
    }  
     
    public void RemoveBeforeMaxValue()  
    {  
    if (IsEmpty())  
    {  
    return;  
    }  
     
    Node current = head;  
    Node maxNode = head;  
     
    while (current != null)  
    {  
    if (current.Data > maxNode.Data)  
    {  
    maxNode = current;  
    }  
    current = current.Next;  
    }  
     
    if (maxNode == head) return;  
     
    current = head;  
    while (current != maxNode)  
    {  
    Node next = current.Next;  
    current.Prev = null;  
    current.Next = null;  
    current = next;  
    }  
     
    head = maxNode;  
    head.Prev = null;  
     
    if (head.Next == null)  
    {  
    tail = head;  
    }  
    }  
     
    IEnumerator IEnumerable.GetEnumerator()  
    {  
    Node current = head;  
    while (current != null)  
    {  
    yield return current.Data;  
    current = current.Next;  
    }  
    }  
     
    public static void PrintList(DoublyLinkedListDouble list, string prompt)  
    {  
    Console.Write($"{prompt}: ");  
    foreach (double item in list)  
    {  
    Console.Write($"{item}, ");  
    }  
    Console.WriteLine();  
    }  
    }  
    }
3. **Node.cs:**
4. namespace DoubleLinkedList;  
     
   public class Node  
   {  
    public double Data;   
      
    public Node Next { get; set; }   
      
    public Node Prev { get; set; }  
     
    public Node(double element)  
    {  
    Data = element;  
    Prev = Next = null;   
    }  
   }

**Program.cs:**

namespace DoubleLinkedList  
{  
 public class Program  
 {  
 public static void Main()  
 {  
 var testList = new DoublyLinkedListDouble();  
  
 testList.AddElementToStart(20);  
 testList.AddElementToStart(35);  
 testList.AddElementToStart(40);  
 testList.AddElementToStart(-85);  
 testList.AddElementToStart(-100);  
 testList.AddElementToStart(500);  
 testList.AddElementToStart(28);  
 testList.AddElementToStart(-28);  
  
 DoublyLinkedListDouble.PrintList(testList, "Added elements to start");  
  
 Console.WriteLine($"The first positive element = {testList.FindFirstPositiveElement()}");  
   
 Console.ForegroundColor = ConsoleColor.*Green*;  
 var avg = testList.FindMeanValue();  
 Console.WriteLine($"Average value = {avg}");  
  
 var max = testList.FindMaxValue();  
 Console.WriteLine($"Maximum value = {max}");  
 Console.ForegroundColor = ConsoleColor.*White*;  
  
 Console.WriteLine($"Number of values bigger than the average = {testList.FindNumberElementsBiggerMeanValue()}");  
 var valuesBiggerMeanValue = testList.GetElementsBiggerMeanValue();  
 Console.WriteLine($"Values bigger than the average: {string.Join(", ", valuesBiggerMeanValue)}");  
  
 testList.RemoveBeforeMaxValue();  
 DoublyLinkedListDouble.PrintList(testList, "List after removing elements before max value");  
 }  
 }  
}