

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

26 август 2022 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 2

Ключ с верните отговори

Задача от 1. до 16.

Въпрос №	Верен отговор	Брой точки
1.	А	1
2.	Б	1
3.	В	1
4.	В	1
5.	Б	1
6.	Б	1
7.	Г	1
8.	А	1
9.	Б	1
10.	Г	1
11.	А	1
12.	В	1
13.	Г	1
14.	А	1
15.	Б	1
16.	В	1

Задача 17. – 3 точки

Верен отговор	$n < 1 \parallel n > 10$
---------------	--------------------------

Задача 18. – 3 точки

Верен отговор	1234 123 12 1
---------------	--

Задача 19. – 3 точки

Верен отговор	<pre>C#: (int i = arr.Length - 1; i >= 0; i--) Java: (int i = arr.length - 1; i >= 0; i--)</pre>
---------------	--

Задача 20. – 3 точки

Верен отговор	<pre>SELECT city, COUNT(*) FROM schools GROUP BY city;</pre>
---------------	--

Задача 21. – 3 точки

Верен отговор	<pre>C# string [] word = sentence.Split(new char[]{' '}); или string [] word = sentence.Split(); или string [] word = sentence.Split(" ");</pre>
	<pre>Java- String[] word = sentence.split(" "); или String[] word = sentence.split(" ",5); или String[] word = sentence.split("\\s"); или String[] word = sentence.split("\\s+");</pre>

Задача 22. – 3 точки

Верен отговор	-1;3;
---------------	-------

Задача 23. – 3 точки

Верен отговор	product_id	max_price
	5	160
	20	140

Задача 24. – 3 точки

Верен отговор	C#
	<pre>static int[] BubbleSort (int[] list) { for (int i = 0 ; i < list.Length-1 ; i++) for (int j = 0 ; j < list.Length-1-i ; j++) if (list[j] < list[j + 1]) { int temp = list[j]; list[j] = list[j + 1]; list[j + 1] = temp; } return list; }</pre>
	JAVA
	<pre>static int[] bubbleSort(int[] array){ for (int i = 0; i < array.length-1; i++){ for (int j = 0; j < array.length-i-1; j++) if (array[j] < array[j + 1]) { int temp = array[j]; array[j] = array[j + 1]; array[j + 1] = temp; } } return array; }</pre>

Забележка: Признават се и всички други верни отговори и предложени решения.

Задача 25. – 10 точки

Примерно решение:

C#
<pre>using System; namespace zad25 { class Program { static void Main (string[] args) { try { Console.WriteLine ("Въведете коефициента a:"); int a = int.Parse (Console.ReadLine ()); } } } }</pre>

```

        Console.WriteLine ("Въведете коефициента b:");
        int b = int.Parse (Console.ReadLine ());
        if ( b <= 0 )
        {
            if ( a >= 0 )
                Console.WriteLine ("Няма реални решения");
            else
            {
                double root = Math.Sqrt (b / a);
                Console.WriteLine ($"Решенията са (-inf; {-
root:f2}) U ({root:f2}; +inf)");
            }
        }
        else
        {
            if ( a <= 0 )
                Console.WriteLine ("Всички реални числа са
решения");
            else
            {
                double root = Math.Sqrt (b / a);
                Console.WriteLine ($"Решенията са ({-
root:f2}; {root:f2})");
            }
        }
    }
    catch ( FormatException )
    {
        Console.WriteLine ("Некоректно въведено число");
    }
}
}
}

```

Java

```

import java.util.Scanner;

public class zad25 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.println("Въведете коефициента a:");
            int a = Integer.parseInt(scanner.nextLine());
            System.out.println("Въведете коефициента b:");

```

```

        int b = Integer.parseInt(scanner.nextLine());
        if (b <= 0)
            if (a >= 0)
                System.out.println("Няма реални решения");
            else {
                double root = Math.sqrt(1.0*b/a);
                System.out.println("Решенията са (-inf; " +
String.format("%.2f", -root) + ") U (" + String.format("%.2f", root)
+ "; +inf)");
            }
        else
            if (a <= 0)
                System.out.println("Всички реални числа са
решения");
            else {
                double root = Math.sqrt(1.0*b/a);
                System.out.println("Решенията са (" +
String.format("%.2f", -root) + "; " + String.format("%.2f", root) +
")");
            }
    } catch (NumberFormatException e) {
        System.out.println("Некоректно въведено число");
    }
}
}
}

```

Задача 26 – 15 точки

Примерно решение:

```

C#

class Program
{
    static void Main ( string[] args )
    {
        List<Human> lstHuman = new List<Human> ();
        string firstName, lastname;
        int age;
        Console.Write("n=");
        int n = int.Parse (Console.ReadLine ());
        for ( int i = 0 ; i < n ; i++ )
        {

```

```

        Console.WriteLine ("First name: ");
        firstName = Console.ReadLine ();
        Console.WriteLine ("Last name: ");
        lastname = Console.ReadLine ();
        Console.WriteLine ("Age: ");
        age = int.Parse (Console.ReadLine ());
        Console.WriteLine ("Your choice[s - student] , [w -
worker]: ");

        string choice = Console.ReadLine();
        if ( choice == "s" )
        {
            Console.WriteLine ("Grade: ");
            double mark = double.Parse (Console.ReadLine ());
            lstHuman.Add (new Student (firstName , lastname ,
age , mark));
        }
        else if ( choice == "w" )
        {
            Console.WriteLine ("Wage: ");
            double wage = double.Parse (Console.ReadLine ());
            Console.WriteLine ("Hours worked: ");
            double workHours = double.Parse (Console.ReadLine
());
            lstHuman.Add (new Worker (firstName , lastname ,
age , wage , workHours));
        }
    }

    for ( int i = lstHuman.Count - 1 ; i >= 0 ; i-- )
    {
        Console.WriteLine (lstHuman[i]);
    }
}

public class Human
{
    private string firstName;
    private string lastName;
    private int age;

    public Human ( string firstName , string lastName , int age )
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
    }
}

```

```

    }
    public override string ToString ()
    {
        return this.firstName + " " + this.lastName + ", " +
this.age + " years old";
    }
}

public class Student : Human
{
    private double mark;

    public Student ( string firstName , string lastName , int
age , double mark ) : base (firstName , lastName , age) { this.mark =
mark; }

    public override string ToString ()
    {
        return $"{base.ToString ()}, grade: {this.mark:f2}";
    }
}
public class Worker : Human
{
    private double wage;
    private double workHours;

    public Worker ( string firstName , string lastName , int
age , double wage , double workHours ) : base (firstName , lastName ,
age)
    {
        this.wage = wage;
        this.workHours = workHours;
    }

    public double WagePerHour ()
    {
        return this.wage * this.workHours;
    }

    public override string ToString ()
    {
        return $"{base.ToString ()}, salary: ${WagePerHour
():f2}";
    }
}

```

```
}
```

Java

```
package zad26;
```

```
public class Human {
```

```
    private final String firstName;
```

```
    private final String lastName;
```

```
    private final Integer age;
```

```
    public Human(String firstName, String lastName, Integer age) {
```

```
        this.firstName = firstName;
```

```
        this.lastName = lastName;
```

```
        this.age = age;
```

```
    }
```

```
    @Override
```

```
    public String toString() {
```

```
        return this.firstName + " " + this.lastName + ", " + this.age  
+ " years old";
```

```
    }
```

```
}
```

```
package zad26;
```

```
public class Worker extends Human {
```

```
    private final Double wage;
```

```
    private final Integer workHours;
```

```
    public Worker(String firstName, String lastName, Integer age,  
Double wage, Integer workHours) {
```

```
        super(firstName, lastName, age);
```

```
        this.wage = wage;
```

```
        this.workHours = workHours;
```

```
    }
```

```
    public double Salary() {
```

```
        return this.wage * workHours;
```

```
    }
```

```
    @Override
```

```
    public String toString() {
```

```
        return super.toString() + ", salary: " +  
String.format("$%,.2f", Salary());
```

```
    }
```



```

}
package zad26;

public class Student extends Human {

    private final double mark;

    public Student(String firstName, String lastName, Integer age,
Double mark) {
        super(firstName, lastName, age);
        this.mark = mark;
    }

    @Override
    public String toString() {
        return super.toString() + ", grade: " +
String.format("%.2f",this.mark);
    }
}
package zad26;

import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;

public class zad26 {

    public static void main(String[] args) {

        try (Scanner scanner = new Scanner(System.in)) {
            List<Human> lstHuman = new LinkedList<>();
            String firstName, lastName;
            Integer age;
            String typeObject;
            System.out.println("n= ");
            Integer n = scanner.nextInt();
            for (int i = 0; i < n; i++) {

                System.out.println("First name: ");
                firstName = scanner.next();
                System.out.println("Last name: ");
                lastName = scanner.next();
                System.out.println("Age: ");
                age = scanner.nextInt();
                System.out.print("Your choice [s-student], [w-worker]: ");
                typeObject = scanner.next();
            }
        }
    }
}

```



```
INSERT INTO students ( ID, Име, БЕЛ, English, Math, Informatika, IT )
VALUES (2, 'Асен Ангелов', 5, 5, 6, 4, 5);
INSERT INTO students ( ID, Име, БЕЛ, English, Math, Informatika, IT )
VALUES (3, 'Борислав Ганев', 4, 4, 5, 5, 6);
INSERT INTO students ( ID, Име, БЕЛ, English, Math, Informatika, IT )
VALUES (4, 'Бояна Тодорова', 5, 6, 6, 6, 6);
INSERT INTO students ( ID, Име, БЕЛ, English, Math, Informatika, IT )
VALUES (5, 'Валери Илиев', 6, 6, 6, 6, 6);
```

Заявка 7:

```
SELECT * FROM students
WHERE ID=4;
```

Заявка 8:

students.ID може да бъде и students.Име. Текстът в квадратни скоби е пояснителен, може да бъде формулиран и по друг начин. Ако е само една дума, скобите могат да се пропуснат

```
SELECT Count(students.ID) AS [Брой ученици]
FROM students
WHERE (((students.Math)=6) AND ((students.Informatika)=6) AND
((students.IT)=6));
```

или

```
SELECT Count(students.ID) AS [Брой ученици]
FROM students
WHERE students.Math=6 AND students.Informatika=6 AND students.IT=6;
```

Заявка 9.

```
SELECT Avg(students.BEL) AS [СУ по БЕЛ], Avg(students.Math) AS [СУ по
Математика]
FROM students;
```

Заявка 10.

```
SELECT students.Име, (students.BEL + students.English +
students.Informatika + students.IT + students.Math)/5 AS SU
FROM students
ORDER BY (students.BEL + students.English + students.Informatika +
students.IT + students.Math)/5 DESC, students.Име;
```

Примерно решение за SQL Server:

Заявка 1.

Аналогична на MS Access

или

```
CREATE TABLE students(
ID int NOT NULL PRIMARY KEY,
```

```
Ime nvarchar(100),
BEL tinyint,
English tinyint,
Math tinyint,
Informatika tinyint,
IT tinyint
);
```

Заявки 2-6.

```
INSERT INTO students ( ID, Ime, BEL, English, Math, Informatika, IT )
VALUES (11, N'Антония Колева', 4, 5, 6, 5, 4);
INSERT INTO students ( ID, Ime, BEL, English, Math, Informatika, IT )
VALUES (21, N'Асен Ангелов', 5, 5, 6, 4, 5);
INSERT INTO students ( ID, Ime, BEL, English, Math, Informatika, IT )
VALUES (31, N'Борислав Ганев', 4, 4, 5, 5, 6);
INSERT INTO students ( ID, Ime, BEL, English, Math, Informatika, IT )
VALUES (41, N'Бояна Тодорова', 5, 6, 6, 6, 6);
INSERT INTO students ( ID, Ime, BEL, English, Math, Informatika, IT )
VALUES (51, N'Валери Илиев', 6, 6, 6, 6, 6);
```

Заявка 7.

Аналогична на MS Access

Заявка 8.

Аналогична на MS Access

Заявка 9.

Аналогична на MS Access

Заявка 10.

```
SELECT students.Ime, (students.BEL + students.English +
students.Informatika + students.IT + students.Math)/5.0 AS SU
FROM students
ORDER BY (students.BEL + students.English + students.Informatika +
students.IT + students.Math)/5.0 DESC, students.Ime;
```

Задача 28 – 20 точки

Примерно решение:

C#

```
using System;
using System.Collections.Generic;
using System.IO;
```

```

namespace Task28
{
    public class OverlappingScopeTest
    {
        private static bool CheckRow ( char[] row , int startCol ,
string word )
        {
            if ( row.Length - startCol < word.Length )
                return false;

            for ( int i = 0 ; i < word.Length ; ++i )
            {
                if ( row[i + startCol] != word[i] )
                    return false;
            }
            return true;
        }
        private static bool Contains ( char[][] table , string word )
        {
            if ( table == null || word == null )
                return false;
            if ( word.Length == 0 )
                return true;

            word = word.ToLower ();
            char[] chars = word.ToCharArray ();
            Array.Reverse (chars);
            string revWord = new string (chars);

            for ( int row = 0 ; row < table.Length ; ++row )
            {
                for ( int col = 0 ; col < table[row].Length ; ++col )
                {
                    if ( CheckRow (table[row] , col , word) ||
CheckRow (table[row] , col , revWord) ) return true;
                }
            }
            return false;
        }
        private static char[][] ReadMatrix ( string fileName )
        {
            StreamReader file;
            try
            {
                file = File.OpenText (fileName);
            }
        }
    }
}

```

```

    }
    catch ( IOException ex )
    {
        Console.WriteLine ("Cannot access file {0} : {1}." ,
fileName , ex.Message);
        return null;
    }

    List<char[]> result = new List<char[]> ();
    try
    {
        for ( string line = file.ReadLine () ; line != null ;
line = file.ReadLine () )
        {
            if ( line.Length > 0 )
                result.Add (line.ToCharArray ());
            if ( result.Count > 1 && result[result.Count -
1].Length != result[result.Count - 2].Length )
            {
                Console.WriteLine ("Invalid format: Matrix is
not rectangular!!");
                return null;
            }
        }
    }
    finally
    {
        file.Close ();
    }
    return result.ToArray ();
}
private static string[] ReadWords ( string fileName )
{
    StreamReader file;
    try
    {
        file = File.OpenText (fileName);
    }
    catch ( IOException ex )
    {
        Console.WriteLine ("Cannot access file {0} : {1}." ,
fileName , ex.Message);
        return null;
    }

    string allLines = file.ReadToEnd ();

```

```

        file.Close ();
        return allLines.Split ("\r\n".ToCharArray () ,
StringSplitOptions.RemoveEmptyEntries);
    }
    static void Main ()
    {
        try
        {
            Console.WriteLine ("Name of the file with the
table:");

            String tableFileName = Console.ReadLine ();
            char[][] table = ReadMatrix (tableFileName);
            if ( table == null ) return;
            Console.WriteLine ("Name of the file with the
words:");

            String wordsFileName = Console.ReadLine ();
            string[] words = ReadWords (wordsFileName);

            if ( words == null ) return;

            foreach ( string word in words )
                if ( Contains (table , word) )
                    Console.WriteLine (word);
        }
        catch ( Exception ex )
        {

            Console.WriteLine (ex.Message);
        }
    }
}

```

Java

```

package task28;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

public class task28 {

    public static char[][] readTable(String fileName) {

```

```

        Scanner scanner;
        try {
            scanner = new Scanner(new File(fileName));
        } catch (IOException ex) {
            System.err.println("Cannot access file: " +
fileName);
            System.err.println(ex.getMessage());
            return null;
        }

        ArrayList<char[]> table = new ArrayList<>();
        int last = -1;
        try {
            while (scanner.hasNext()) {
                char[] line = scanner.nextLine().toCharArray();
                if (last >= 0 && last != line.length) {
                    System.err.println("Invalid format:
Matrix is not rectangular!!");
                    return null;
                }
                last = line.length;
                table.add(line);
            }
        } finally {
            scanner.close();
        }
        return table.toArray(new char[0][0]);
    }

    public static String[] readWords(String fileName) {
        Scanner scanner;
        try {
            scanner = new Scanner(new File(fileName));
        } catch (IOException ex) {
            System.err.println("Cannot access file: " +
fileName);
            System.err.println(ex.getMessage());
            return null;
        }

        ArrayList<String> words = new ArrayList<>();
        try {
            while (scanner.hasNext()) {
                String line = scanner.nextLine();
                if (line.length() > 0) {
                    words.add(line);
                }
            }
        }
    }

```



```

        }
    } finally {
        scanner.close();
    }
    return words.toArray(new String[0]);
}

public static Boolean matchRow(char[] row, int pos, String
word) {
    if (row.length - pos < word.length()) {
        return false;
    }
    for (int i = 0; i < word.length(); ++i) {
        if (row[pos + i] != word.charAt(i)) {
            return false;
        }
    }
    return true;
}

public static Boolean contains(char[][] table, String word) {
    if (word == null || table == null || table.length == 0)
        return false;
    if (word.length() == 0)
        return true;

    // word = word.toLowerCase();
    String revWord = new
StringBuilder(word).reverse().toString();

    for (int row = 0; row < table.length; ++row) {
        for (int col = 0; col < table[row].length; ++col) {
            if (matchRow(table[row], col, word) ||
matchRow(table[row], col, revWord))
                return true;
        }
    }
    return false;
}

public static void main(String args[]) {
    try (Scanner sc = new Scanner(System.in)) {
        System.out.println("Name of the file with the table:
");
        String tableFileName = sc.nextLine();

```

```

        char[][] table = readTable(tableFileName);
        if (table == null) {
            return;
        }
        System.out.println("Name of the file with the
words:");
        String wordsFileName = sc.nextLine();
        String[] words = readWords(wordsFileName);

        if (words == null) {
            return;
        }
        for (String word : words) {
            if (contains(table, word))
                System.out.println(word);
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

```

Забележка: Признават се и всички други верни отговори и предложени решения.