

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

25 август 2023 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 2

Задача от 1. до 16. Ключ с верните отговори

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

Задача 17. – 3 точки

(1) $j < i$

(2) $arr[i, j] < 0$ (за C#)

$arr[i][j] < 0$ (за Java)

или

(1) $j < m$

(2) $arr[i, j] < 0 \ \&\& \ j < i$ (за C#)

$arr[i][j] < 0 \ \&\& \ j < i$ (за Java)

(3) $cnt++$; или $++cnt$; или $cnt += 1$; или $cnt = cnt + 1$;

Задача 18. – 3 точки

5

Задача 19. – 3 точки

(1) има (has a)

(2) е (is a)

(3) има (has a)

Задача 20. – 3 точки

1) INSERT INTO Fee

VALUES

Или

INSERT INTO ... VALUES ...

2) UPDATE Fee SET ...

WHERE ...

Или

UPDATE ... SET ... WHERE ...

3) DELETE FROM Fee

WHERE ...

Или

DELETE FROM ... WHERE ...

Задача 21. – 3 точки

Методът връща като резултат броят четни цифри на положителните числата от интервала [a ; b]. Ако a и b са цели неположителни числа, методът връща резултат нула.

Задача 22. – 3 точки

cars_count	first_year	total_price
4	2000	70000

Задача 23. – 3 точки

информатика 2023

Държавен изпит по информатика 2023

Държавен

зрелостен

изпит

по

информатика

2023

Задача 24. – 3 точки

ред 3 – грешен синтаксис за взимане на дължината на масива, посочено е като метод, вместо като свойство. Правилният код за реда е:

за C# `for (int i = 1; i < a.Length; i++)`

за Java `for (int i = 1; i < a.length; i++)`

ред 4 – грешна проверка дали `a[i]` се дели на `i`. Правилният код за реда е:

`if (a[i] % i == 0)`

ред 5 – липсва оператор за присвояване. Правилният код за реда е:

`s += a[i];` или `s = s + a[i];`

Задача 25. – 10 точки

Примерно решение

C#

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

namespace zad25
{
    internal class Program
    {
        static void Main()
        {
            int number;
            int originalNumber;
            int n; // дължината на списъка
            bool check = true; // проверка за палиндром
            List<int> digits = new List<int>();

            try
            {
                number = int.Parse(Console.ReadLine());
                //валидация
                if (number < 1 || number > 10000000) throw new Exception();

                originalNumber = number;
                while (number > 0) // отделяне на цифрите
                {
                    digits.Add(number % 10);
                    number /= 10;
                }

                n = digits.Count;
                for (int i = 0 ; i < n / 2 ; i++) // проверка за палиндром
                {
                    if (digits[i] != digits[n - i - 1])
                    {
                        check = false;
                        break;
                    }
                }

                if (check)
                {
                    Console.WriteLine(originalNumber + " is a palindrome");
                }
                else
                {
                    Console.WriteLine(originalNumber + " is NOT a palindrome");
                }
            }
            catch (Exception )
            {
                Console.Error.WriteLine("Incorrectly entered number");
            }
        }
    }
}
```

Java

```

package dzi;

import java.util.ArrayList;
import java.util.Scanner;

public class zad25 {

    public static void main(String[] args) {
        int number;
        int originalNumber;
        int n; // дължината на списъка
        boolean check = true; // отметка за палиндром
        ArrayList<Integer> digits = new ArrayList<>();
        try (Scanner input = new Scanner(System.in)) {
            try {
                number = input.nextInt();
                // валидация
                if (number < 1 || number > 10000000)
                    throw new Exception();
                originalNumber = number;
                // отделяне на цифрите и записване в списък
                while (number > 0) {
                    digits.add(number % 10);
                    number /= 10;
                }

                n = digits.size();
                for (int i = 0; i < n / 2; i++) { // проверка за палиндром
                    if (!digits.get(i).equals(digits.get(n - 1 - i))) {
                        check = false;
                        break;
                    }
                }

                if (check) {
                    System.out.printf("%d is a palindrome\n",
originalNumber);
                } else {
                    System.out.printf("%d is NOT a palindrome\n",
originalNumber);
                }
            } catch (Exception ex) {
                System.err.println("Incorrectly entered number");
            }
        }
    }
}

```

Задача 26 – 15 точки

Примерно решение

C#

```

using System;
using System.Collections.Generic;

namespace zad26
{
    internal class Program
    {
        static List<double> ReadPoints()

```

```

{
    List<double> points = new List<double>();
    int n = int.Parse(Console.ReadLine());
    double point;
    for (int i = 0 ; i < n ; i++)
    {
        point = double.Parse(Console.ReadLine());
        if (point > 0)
        {
            points.Add(point);
        }
    }
    return points;
}
static double minDpoints(List<double> points)
{
    double minD = 101;
    for (int i = 1 ; i < points.Count ; i++)
    {
        if (points[i].Equals(points[i - 1])) continue;
        double d = points[i - 1] - points[i];
        if (d < minD) minD = d;
    }
    return minD;
}
static int Laureates(List<double> points)
{
    int countLaureates = 1, cntTo3 = 1;
    for (int i = 1 ; i < points.Count ; i++)
    {
        if (points[i] != points[i - 1])
        {
            cntTo3++;
            if (cntTo3 > 3) break;
        }
        countLaureates++;
    }
    return countLaureates;
}
static void Main()
{
    // прочитане на точките и записване в колекция на валидните работи
    List<double> points = ReadPoints();

    //сортиране на данните в намаляващ ред
    points.Sort((a, b) => b.CompareTo(a));

    // минимална разлика
    double d = minDpoints(points);

    // намиране на броя на лауреатите
    int cntLaureates = Laureates(points);

    // извеждане на резултатите
    Console.WriteLine($"valid works - {points.Count}");
    Console.WriteLine($"minimal difference - {d:F3} p.");
    Console.WriteLine($"laureates - {cntLaureates}");
}
}
}

```

Java

```
package dzi;
```

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;
```

```

public class zad26 {

    static ArrayList<Double> readPoints() {
        ArrayList<Double> points = new ArrayList<>();
        Scanner sc = new Scanner(System.in);
        int n = Integer.parseInt(sc.nextLine());
        double point;
        for (int i = 0; i < n; i++) {
            point = Double.parseDouble(sc.nextLine());
            if (point > 0) {
                points.add(point);
            }
        }
        return points;
    }

    static int Laureates(ArrayList<Double> all) {
        int countAll = 1, cntTo3 = 1;
        for (int i = 1; i < all.size(); i++) {
            if (!(all.get(i).equals(all.get(i - 1)))) {
                cntTo3++;
                if (cntTo3 > 3)
                    break;
            }
            countAll++;
        }
        return countAll;
    }

    static double minDpoints(ArrayList<Double> li) {
        double minD = 101;
        double td;
        for (int i = 1; i < li.size(); i++) {
            if (li.get(i).equals(li.get(i - 1)))
                continue;
            td = li.get(i - 1) - li.get(i);
            if (td < minD) {
                minD = td;
            }
        }
        return minD;
    }

    public static void main(String[] args) {
        // прочитане на поредицата от числа
        ArrayList<Double> points = readPoints();

        // сортиране на данните в намаляващ ред
        Collections.sort(points);
        Collections.reverse(points);

        // минимална разлика
        double d = minDpoints(points);

        // намиране на броя на лауреатите
        int cntLaureates = Laureates(points);

        // извеждане на резултатите
        System.out.println("valid works - " + points.size());
        System.out.println(String.format("minimal difference - %.3f p.", d));
        System.out.println("laureates - " + cntLaureates);
    }
}

```

```
}  
}
```

Задача 27 – 15 точки

Примерно решение

```
-- А)  
CREATE DATABASE music;  
  
CREATE TABLE singers(  
    id INT PRIMARY KEY,  
    name NVARCHAR(100),  
    songs INT,  
    rank INT,  
    networth INT  
);  
  
-- Б)  
INSERT INTO singers(id, name, songs, rank, networth)  
VALUES (1, 'Ivan Ivanov', 50, 1, 1000000),  
(2, 'Maria Ivanova', 45, 3, 900000),  
(3, 'Georgi Georgiev', 20, 4, 800000),  
(4, 'Gergana Petrova', 55, 2, 1000000),  
(5, 'Boris Borisov', 35, 5, 900000);  
  
-- В)  
SELECT rank, name  
FROM singers  
WHERE rank <= 3  
ORDER BY rank;  
  
rank      name  
-----  
1         Ivan Ivanov  
2         Gergana Petrova  
3         Maria Ivanova  
  
-- Г)  
SELECT COUNT(*), AVG(networth) / 1.95583  
FROM singers;  
  
-----  
5         470388.5307005  
  
-- Д)  
UPDATE singers  
SET networth = networth * 1.1  
WHERE ID BETWEEN 2 AND 4
```

След изпълнение на заявката таблицата ще има следното съдържание:

```
-----  
1         Ivan Ivanov          50         1         1000000  
2         Maria Ivanova       55         3         990000  
3         Georgi Georgiev     20         4         880000  
4         Gergana Petrova     55         2         1100000  
5         Boris Borisov       20         5         900000
```


Задача 28 – 20 точки

Примерно решение

C#

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

namespace zad28
{
    class Pilot
    {
        private string name;
        private int age;
        private string category;
        private Car carp;
        public Pilot(string name, int age, string categ, Car car)
        {
            this.name = name;
            this.age = age;
            this.category = categ;
            this.carp = car;
        }
        public override string ToString()
        {
            return $"{name}, {age}, {category}, {carp}";
        }
    }

    class Car
    {
        private string model;
        private int hPower;
        public Car(string model, int hPower)
        {
            this.model = model;
            this.hPower = hPower;
        }
        public override string ToString()
        {
            return $"{model}, {hPower}";
        }
    }

    class Rally
    {
        private string rName;
        private int rYear;
        private List<Pilot> pilots;
        public Rally(string name, int year)
        {
            this.rName = name;
            this.rYear = year;
            pilots = new List<Pilot>();
        }
        public void AddPilot()
        {
            Console.Write("Pilot name: ");
            string name = Console.ReadLine();
            Console.Write("Age: ");
            int age = int.Parse(Console.ReadLine());
            Console.Write("Category (A|B|C): ");
            string categ = Console.ReadLine();
            Console.Write("Car model: ");
        }
    }
}
```

```

        string model = Console.ReadLine();
        Console.Write("Car power (hp): ");
        int hpower = int.Parse(Console.ReadLine());
        Pilot pilot = new Pilot(name, age, categ, new Car(model, hpower));
        pilots.Add(pilot);
    }

    public void ViewAll()
    {
        Console.WriteLine($"Rally: {rName} - {rYear}");
        Console.WriteLine(string.Join(Environment.NewLine, pilots));
    }
}

class Zad28
{
    static void Main()
    {
        string rName = Console.ReadLine();
        int rYear = int.Parse(Console.ReadLine());
        Rally rally = new Rally(rName, rYear);
        while (true)
        {
            string inp;
            do
            {
                Console.WriteLine();
                Console.WriteLine("[a]dd [v]iew [q]uit");
                inp = Console.ReadLine();
            } while (inp != "a" && inp != "v" && inp != "q");

            if (inp == "q") { break; }

            switch (inp)
            {
                case "a":
                    rally.AddPilot();
                    break;
                case "v":
                    rally.ViewAll();
                    break;
            }
        }
    }
}

```

Java

```

package dzi;

import java.util.ArrayList;
import java.util.Scanner;

public class zad28 {

    private static Scanner input;

    public static void main(String[] args) {
        input = new Scanner(System.in);
        String name = input.nextLine();
        int year = Integer.parseInt(input.nextLine());

        Rally rally = new Rally(name, year);

        while (true) {

```

```

String inp;
do {
    System.out.println();
    System.out.println("[a]dd [v]iew [q]uit");
    inp = input.nextLine();
} while (!inp.equals("a") && !inp.equals("v") && !inp.equals("q"));

if (inp.equals("q")) {
    break;
}
switch (inp) {
case "a":
    rally.addPilot();
    break;
case "v":
    rally.ViewAll();
    break;
}
}
}

class Pilot {
    private String name;
    private int age;
    private Car carp;
    private String category;

    public Pilot(String name, int age, Car carp, String category) {
        this.name = name;
        this.age = age;
        this.carp = carp;
        this.category = category;
    }

    @Override
    public String toString() {
        return this.name + "," + this.age + "," + this.category + "," + this.carp;
    }
}

class Car {

    private String brand;
    private int hPower;

    public Car(String brand, int hPower) {
        this.brand = brand;
        this.hPower = hPower;
    }

    @Override
    public String toString() {
        return String.format("%s,%d", brand, hPower);
    }
}

class Rally {

    private String rName;
    private int rYear;
    private ArrayList<Pilot> pilots;

```

```

public Rally(String name, int year) {
    this.rName = name;
    this.rYear = year;
    this.pilots = new ArrayList<Pilot>();
}

public void addPilot() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Pilot name: ");
    String name = sc.nextLine();
    System.out.print("Age: ");
    int age = Integer.parseInt(sc.nextLine());
    System.out.print("Category (A|B|C): ");
    String categ = sc.nextLine();
    System.out.print("Car model: ");
    String model = sc.nextLine();
    System.out.print("Car power (hp): ");
    int hpower = sc.nextInt();
    Pilot pilot = new Pilot(name, age, new Car(model, hpower), categ);
    pilots.add(pilot);
}

public void ViewAll() {
    System.out.println("Rally: " + this.rName + " - " + this.rYear);
    for (Pilot pilot : pilots) {
        System.out.println(pilot);
    }
}
}

```