

【7】 次の各問いに答えなさい。

問1. プログラムの説明を読んで、プログラムの(1)~(2)を答えなさい。

<プログラムの説明>

処理内容

ある会社の部署対抗ボウリング大会のデータを入力し、集計した結果をディスプレイに表示する。

入力データ

部署コード ×	スコア ×××
------------	------------

(第1図)

実行結果

部署コードを入力 =>1		
スコアを入力 =>80		
部署コードを入力 =>2		
スコアを入力 =>120		
部署コードを入力 =>1		
スコアを入力 =>203		
部署コードを入力 =>2		
スコアを入力 =>198		
}		
部署コードを入力 =>2		
スコアを入力 =>108		
部署コードを入力 =>2		
スコアを入力 =>125		
部署コードを入力 =>-1		
<集計結果>		
	総務部	営業部
[合計スコア]	1,100	1,083
[勝敗判定]	総務部チームの勝利!	
[個人最高スコア]	203	

(第2図)

処理条件

1. キーボードから第1図の部署コードとスコアを入力する。なお、部署コードは1が総務部、2が営業部である。
2. 部署ごとに合計スコアを集計する。
3. キーボードから部署コードに -1 が入力されたら、次の処理を行う。
 - 第2図のように部署ごとの合計スコアを表示する。
 - スコアの大きい部署を勝利チームとして表示する。なお、合計スコアが同じ場合は、「引き分け」を表示する。
 - 最後に、個人最高スコアを表示して処理を終了する。なお、個人最高スコアが複数ある場合は、先に入力したデータを優先する。

<Javaプログラム>

```
//クラスBowling
import java.util.Scanner;

public class Bowling {
    public static void main(String[] args) {
        int code;
        int score;
        int somu = 0;
        int eigyo = 0;
        int saiko = 0;
        Scanner sc = new Scanner(System.in);
        System.out.print("部署コードを入力 =>");
        code = sc.nextInt();
        while(code != -1) {
            System.out.print("スコアを入力 =>");
            score = sc.nextInt();
            if(code == 1) {
                (1);
            } else {
                eigyo = eigyo + score;
            }
            if(score > saiko) {
                saiko = score;
            }
            System.out.print("部署コードを入力 =>");
            code = sc.nextInt();
        }
        sc.close();
        System.out.println();
        System.out.println("<集計結果>");
        System.out.println("                総務部    営業部");
        System.out.printf("[合計スコア]    %4d    %4d\n", somu, eigyo);
        System.out.print("[勝敗判定]    ");
        if(somu > eigyo) {
            System.out.println("総務部チームの勝利!");
        } else if(somu < eigyo) {
            System.out.println("営業部チームの勝利!");
        } else {
            System.out.println("引き分け");
        }
        System.out.printf("[個人最高スコア]    %3d", (2));
    }
}
```

問 2. プログラムの説明を読んで、プログラムの(3)~(5)を答えなさい。

<プログラムの説明>

処理内容

自動車メーカーの走行データを読み、車種別燃費調査結果をディスプレイに表示する。

入力データ

走行データ (ファイル名: drive.csv)

日	分類コード	車種コード	走行距離	消費量
××	×	××	×××	××

(第 1 図)

実行結果

車種別燃費調査結果				
車種名	走行距離	消費量	燃費	備考
フィット	2,802	150	18	
フリーダム	2,787	160	17	
アコーディオン	2,367	161	14	△
ヴェゼルズ	2,669	177	15	
スローリー	2,611	147	17	
フィットHV	2,879	131	21	◎
フリーダムHV	2,770	160	17	
ヴェゼルズHV	2,661	162	16	
KBOX	2,523	128	19	
KWGN	2,605	119	21	◎
ガソリン車	の平均燃費は		16	
ハイブリッド車	の平均燃費は		18	
軽自動車	の平均燃費は		20	

(第 2 図)

処理条件

- 第 1 図の分類コードは、1 がガソリン車、2 がハイブリッド車、3 が軽自動車である。
- 第 1 図の車種コードは、1 ~ 10 であり、配列 syasyuMei と添字で対応している。

配列

syasyuMei	(0)	(1)	~	(10)
		フィット	~	KWGN

- 第 1 図の走行データを読み、次の処理を行う。
 - 分類コードごとに、走行距離と消費量を集計する。
 - 配列 kyoriSyukei に走行距離を、配列 syohiSyukei に消費量を集計する。なお、各配列は配列 syasyuMei と添字で対応している。

配列

kyoriSyukei	(0)	(1)	~	(10)
			~	
syohiSyukei	(0)	(1)	~	(10)
			~	

- データを読み終えたあと、燃費を求めて第 2 図のように表示する。なお、備考は、燃費が 20 以上の場合「◎」を、15 未満の場合「△」を表示する。
- 最後に、分類ごとの平均燃費を次の計算式で求め、第 2 図のように表示して処理を終了する。

$$\text{平均燃費} = \text{走行距離合計} \div \text{消費量合計}$$

(注) Integer.parseInt(文字列型)

文字列型のデータを数値型(整数)に変換する。

<Javaプログラム>

```
//クラスBunrui
public class Bunrui {
    public String mei;
    public int kyoriKei = 0;
    public int syohiKei = 0;
    public int heikin;
    public Bunrui(String mei) {
        this.mei = mei;
    }
    public void syukei(int kyori, int syohi) {
        kyoriKei = kyoriKei + kyori;
        syohiKei = syohiKei + syohi;
    }
    public void keisan() {
        heikin = (3);
    }
}

//クラスNenpiSyukei
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class NenpiSyukei {
    public static void main(String[] args) throws IOException {
        String[] syasyuMei = { "", "フィット", "~", "KWGN" };
        int[] kyoriSyukei = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        int[] syohiSyukei = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        Bunrui bunG = new Bunrui("ガソリン車");
        Bunrui bunH = new Bunrui("ハイブリッド車");
        Bunrui bunK = new Bunrui("軽自動車");
        BufferedReader fileIn = new BufferedReader(new FileReader("drive.csv"));
        String line;
        while((line = fileIn.readLine()) != null) {
            String[] str = line.split(",");
            int hi = Integer.parseInt(str[0]);
            int bunruiCode = Integer.parseInt(str[1]);
            int syasyuCode = Integer.parseInt(str[2]);
            int kyori = Integer.parseInt(str[3]);
            int syohi = Integer.parseInt(str[4]);
            if(bunruiCode == 1) {
                bunG.syukei(kyori, syohi);
            } else if(bunruiCode == 2) {
                bunH.syukei(kyori, syohi);
            } else {
                bunK.syukei(kyori, syohi);
            }
            kyoriSyukei[syasyuCode] = kyoriSyukei[syasyuCode] + (4);
            syohiSyukei[syasyuCode] = syohiSyukei[syasyuCode] + syohi;
        }
        fileIn.close();
        System.out.println("車種別燃費調査結果");
        System.out.println("車種名 走行距離 消費量 燃費 備考");
        for(int i = 1; i <= 10; i++) {
            int nenpi = kyoriSyukei[i] / syohiSyukei[i];
            String biko;
            if(nenpi >= 20) {
                biko = "◎";
            } else if((5)) {
                biko = "△";
            } else {
                biko = " ";
            }
            System.out.printf("%-7s %4d %3d %2d %1s\n",
                syasyuMei[i], kyoriSyukei[i], syohiSyukei[i], nenpi, biko);
        }
        bunG.keisan();
        System.out.printf("%-7s の平均燃費は %2d\n", bunG.mei, bunG.heikin);
        bunH.keisan();
        System.out.printf("%-7s の平均燃費は %2d\n", bunH.mei, bunH.heikin);
        bunK.keisan();
        System.out.printf("%-7s の平均燃費は %2d", bunK.mei, bunK.heikin);
    }
}
```