

2019年 1 月20日 実施

平成30年度（第60回）
情 報 処 理 検 定 試 験
〈プログラミング部門〉
第 1 級 試 験 問 題

注 意 事 項

1. 監督者の指示があるまで，試験問題に手を触れないでください。
2. 試験問題は 9 ページあります。
3. 解答はすべて解答用紙に記入します。
4. 【1】 【2】 【3】 【4】 【5】 【6】 は共通問題です。
5. 【7】 の問題は J a v a ・ マクロ言語のいずれか一つを選択し，解答用紙の選択言語を ☐ で囲んでください。
6. 電卓などの計算用具は使用できません。
7. 筆記用具などの物品の貸借はできません。
8. 問題用紙の回収については監督者の指示にしたがってください。
9. 制限時間は60分です。

主催 公益財団法人 全国商業高等学校協会

【1】 次の説明文に最も適した答えを解答群から選び、記号で答えなさい。

1. 「信頼性」、「可用性」、「保守性」、「保全性」、「機密性」の英単語の頭文字をとった、コンピュータシステムの評価指標。
2. ネットワーク通信において、受信したデータに含まれる送信元アドレスなどの情報を検査し、そのデータを通過させるか破棄するか判断するセキュリティ機能。
3. インターネットを安全に利用するため、送受信するデータの作成者が本人であることや、そのデータが改ざんされていないことを確認できるしくみ。
4. データ要素に次の要素の参照先情報を持たせることによって、複数のデータが連結されるデータ構造。
5. 開発したシステムが、要件として定義した処理時間や処理能力を満たしているかを検証するテスト。

解答群

- | | | |
|----------------|----------------|--------------|
| ア. O S I 参照モデル | イ. DMZ | ウ. リスト |
| エ. 電子署名 | オ. フェールセーフ | カ. I M A P |
| キ. プロトコル | ク. 性能テスト | ケ. シンクライアント |
| コ. 結合テスト | サ. パケットフィルタリング | シ. R A S I S |

【2】 次のA群の語句に最も関係の深い説明文をB群から選び、記号で答えなさい。

- < A 群 > 1. レスポンスタイム 2. ドライバ 3. D H C P サーバ
 4. スパイラルモデル 5. N A S

< B 群 >

- ア. コンピュータシステムに処理の要求を出してから、すべての実行結果が得られるまでの時間。
 イ. 複数のモジュールから構成されるプログラムを上位モジュールから順にテストする際、下位モジュールの代替となるもの。
 ウ. ネットワークに接続されたコンピュータに、I P アドレスなどの情報を重複しないように自動的に割り当てるためのサーバ。
 エ. 画像や音声などのバイナリデータを、電子メールで送受信するための規格。
 オ. システムを独立性の高いサブシステムごとに分割し、ユーザの要求などを経て、設計・プログラミング・テストの工程を繰り返すシステム開発手法。
 カ. ネットワーク上で利用されるドメイン名とI P アドレスの対応関係を管理するためのサーバ。
 キ. コンピュータシステムに処理の要求を出してから、実行結果の最初の応答が得られるまでの時間。
 ク. システムを開発する際、初期段階から試作品を作成してユーザと確認しながら進めていく開発手法。
 ケ. ファイルサーバの機能を持ち、直接ネットワークに接続して使用する補助記憶装置。
 コ. 複数のモジュールから構成されるプログラムを下位モジュールから順にテストする際、上位モジュールの代替となるもの。

【3】 次の説明文に最も適した答えをア、イ、ウの中から選び、記号で答えなさい。

1. 8 ビットの2進数の01101001と00110101がある。この2つの数値の和を16進数で表したもの。
 ア. 3 4 イ. 5 C ウ. 9 E
2. 外部設計の次の開発工程であり、ソフトウェアのアルゴリズムや入出力データなどを詳細に設計する工程。
 ア. プログラム設計 イ. 内部設計 ウ. 基本設計
3. 同じネットワーク内に接続された個々のコンピュータ機器を識別するためのアドレス。
 ア. ホストアドレス イ. ネットワークアドレス ウ. ポート番号
4. コンピュータにハードウェアを接続する際、O S が自動的に必要な設定を行い、ハードウェアを使用可能にする機能。
 ア. プラグアンドプレイ イ. キャッシュメモリ ウ. ディスクキャッシュ
5. 15,000回転／分のハードディスク装置の平均回転待ち時間は、何ミリ秒か。
 ア. 0.25ミリ秒 イ. 2ミリ秒 ウ. 4ミリ秒

【 4 】 次の各問いに答えなさい。

問 1. 流れ図の説明を読んで、流れ図の(1)～(3)にあてはまる答えを解答群から選び、記号で答えなさい。

<流れ図の説明>

処理内容

配列に記憶した数値を並べ替えてディスプレイに表示する。

処理条件

1. 配列 Kei に数値を記憶する。なお、データ件数は n 件である。

配列

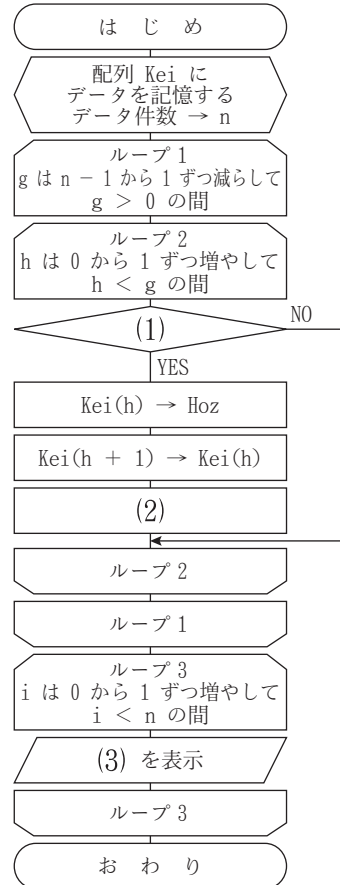
Kei	(0)	(1)	～	($n - 2$)	($n - 1$)
	291	367	～	425	318

2. 配列 Kei の数値を降順に並べ替える。
3. 並べ替えが終わったら、配列 Kei の内容を表示する。

解答群

- ア. $\text{Kei}(h) > \text{Kei}(h + 1)$
 イ. $\text{Kei}(i)$
 ウ. $\text{Hoz} \rightarrow \text{Kei}(h + 1)$
 エ. $\text{Kei}(n)$
 オ. $\text{Hoz} \rightarrow \text{Kei}(h)$
 カ. $\text{Kei}(h) < \text{Kei}(h + 1)$

<流れ図>



問 2. 流れ図の説明を読んで、流れ図の(4)～(5)にあてはまる答えを解答群から選び、記号で答えなさい。

<流れ図の説明>

処理内容

配列に記憶した文字列を探索してメッセージをディスプレイに表示する。

処理条件

1. 配列 Cod に文字列を昇順に記憶する。なお、データ件数は n 件であり、同じ文字列はないものとする。

配列

Cod	(0)	(1)	～	($n - 2$)	($n - 1$)
	B101	B102	～	T104	T105

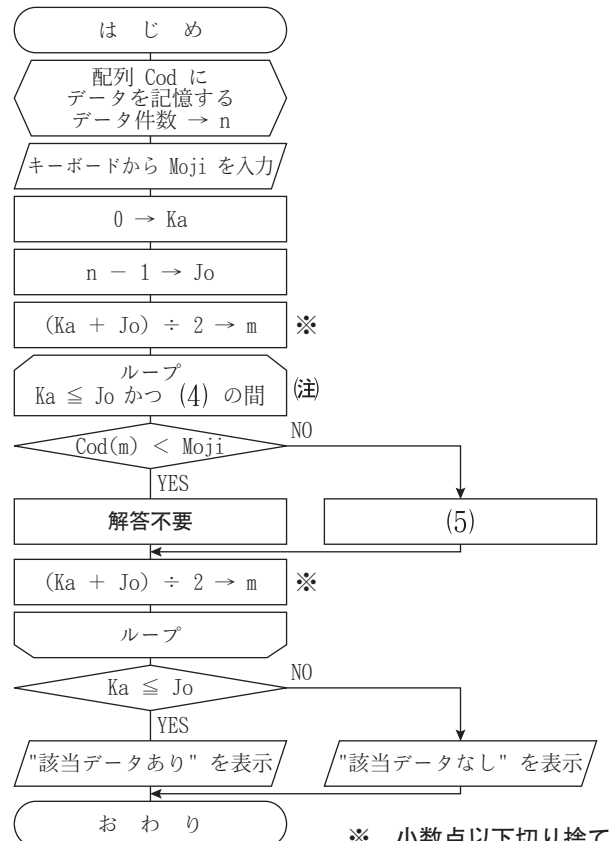
2. キーボードから入力した文字列をもとに配列 Cod を探索し、見つかった場合は 該当データあり を、見つからなかった場合は 該当データなし を表示する。

解答群

- ア. $m - 1 \rightarrow \text{Jo}$
 イ. $\text{Cod}(m) = \text{Moji}$
 ウ. $m + 1 \rightarrow \text{Jo}$
 エ. $\text{Cod}(m) \neq \text{Moji}$

(注) 条件式が「かつ」で複合されている場合、先に記述された条件式が偽になった時点で、判定を終了する。

<流れ図>



※ 小数点以下切り捨て

【 5 】 流れ図の説明を読んで、流れ図の(1)～(5)にあてはまる答えを解答群から選び、記号で答えなさい。

<流れ図の説明>

処理内容

ある企業の工場見学アンケートデータを読み、アンケート集計結果をディスプレイに表示する。

入力データ

日付 (Hiduke)	工場番号 (Kban)	年齢区分 (Nku)	評価点 (Hten)
××××	××	×	×

(第 1 図)

実行結果

	(アンケート集計結果)				
横浜工場	(満足)	(やや満足)	(普通)	(やや不満)	(不満)
(大 人)	42.6%	29.8%	17.0%	8.5%	2.1%
(中高生)	38.9%	30.6%	16.7%	11.1%	2.7%
(小学生)	63.2%	26.3%	5.2%	5.3%	0.0%
(平均評価点)	4.1				
}					
博多工場					
(大 人)	51.0%	25.5%	11.8%	7.8%	3.9%
(中高生)	31.4%	34.2%	18.4%	10.5%	5.5%
(小学生)	9.3%	25.6%	30.2%	30.2%	4.7%
(平均評価点)	3.6				

(第 2 図)

処理条件

- 第 1 図のデータは、工場番号、年齢区分の昇順に記録されている。なお、工場番号は 1 (横浜工場) ～10 (博多工場)、年齢区分は 1 (大人) ～3 (小学生)、評価点は 1 (不満) ～5 (満足) である。また、すべての工場ですべての年齢区分のデータがある。
- 配列 Kmei に工場名を記憶する。なお、Kmei の添字は工場番号と対応している。

配列

Kmei	(0)	(1)	～	(10)
		横浜工場	～	博多工場

- 第 1 図の入力データを読み、次の処理を行う。
 - 工場が変わるごとに工場名を表示する。
 - 配列 Hken に評価別の件数を集計する。なお、Hken(0) には件数計を求める。また、Hken の添字は評価点と対応している。

配列

Hken	(0)	(1)	(2)	(3)	(4)	(5)
	(計)	(不満)	(やや不満)	(普通)	(やや満足)	(満足)

- 年齢区分が変わるごとに合計評価点と合計件数を求める。さらに、各評価の割合を次の計算式で求め、満足から不満までを第 2 図のように表示する。

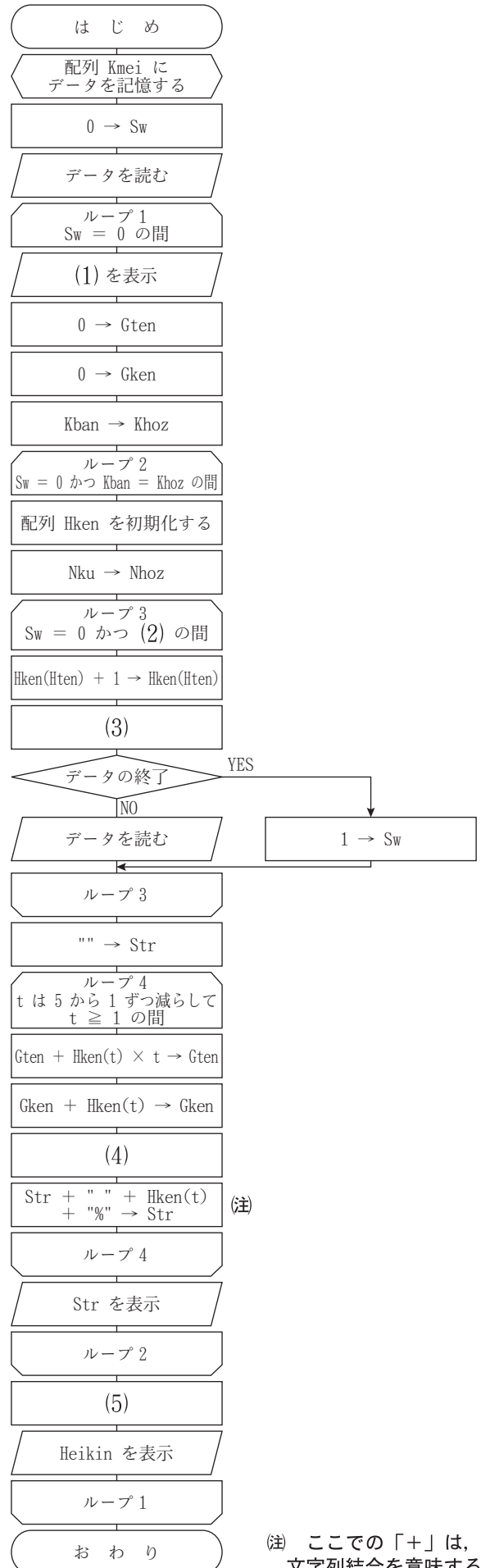
$$\text{各評価の割合} = \text{各評価の件数} \times 100 \div \text{件数計}$$
 - 工場が変わるごとに平均評価点を次の計算式で求め、第 2 図のように表示する。

$$\text{平均評価点} = \text{合計評価点} \div \text{合計件数}$$
- データにエラーはないものとする。

解答群

- ア. $Nku = Nhoz$
 イ. $Hken(t) \times 100 \div Hken(0) \rightarrow Hken(t)$
 ウ. $Gken \div Gten \rightarrow Heikin$
 エ. $Hken(0) + Hken(Hten) \rightarrow Hken(0)$
 オ. $Kban = Nhoz$
 カ. $Kmei(Nku)$
 キ. $Hken(0) + 1 \rightarrow Hken(0)$
 ク. $Hken(0) \times 100 \div Hken(t) \rightarrow Hken(t)$
 ケ. $Kmei(Kban)$
 コ. $Gten \div Gken \rightarrow Heikin$

<流れ図>



(注) ここでの「+」は、
文字列結合を意味する。

【 6 】 流れ図の説明を読んで、流れ図の(1)～(5)にあてはまる答えを解答群から選び、記号で答えなさい。

<流れ図の説明>

処理内容

国内の1年分の農作物収穫データを読み、農作物収穫量統計（単位：トン）をディスプレイに表示する。

入力データ

月 (Tuki)	農作物コード (Ncode)	都道府県コード (Tcode)	収穫量 (トン) (Ryo)
××	×××	××	×～×

(第1図)

実行結果

(農作物収穫量統計 (単位：トン))			
(分類別収穫量計と農作物別収穫量計)			
穀類 9,287,004		(都道府県：収穫量計)	
水陸稲 8,044,438 (収穫量上位)	新潟 : 678,605	北海道 : 578,604	秋田 : 515,400 ~ 岡山 : 162,000
(その他)	埼玉 : 156,610	愛知 : 144,305	三重 : 144,096 ~
}	}	}	}
果樹 2,424,960			
みかん 800,783 (収穫量上位)	和歌山 : 161,100	愛媛 : 127,800	静岡 : 121,295 ~ 佐賀 : 48,804
(その他)	愛知 : 30,501	広島 : 24,096	福岡 : 20,308 ~
}	}	}	}

(第2図)

処理条件

1. 第1図の農作物コードは次の例のように構成されており、分類番号は1(穀類)～4(花き)である。なお、都道府県コードは1～47である。

例 3 0 2 → 3 0 2
分類番号 分類内番号

2. 配列 Bmei に分類名を記憶する。なお、Bmei の添字は分類番号と対応している。

配列

Bmei	(0)	(1)	(2)	(3)	(4)
		穀類	野菜	果樹	花き

3. 配列 Tmei に都道府県名を記憶する。なお、Tmei の添字は都道府県コードと対応している。

配列

Tmei	(0)	(1)	(2)	～	(46)	(47)
		北海道	青森	～	鹿児島	沖縄

4. 配列 Ncd に農作物コードを、配列 Nmei に農作物名を、農作物コードの昇順に記憶する。なお、農作物は60種類であり、Ncd と Nmei の添字は対応している。

配列

Ncd	Nmei
(0)	(0)
(1)	(1)
(2)	(2)
}	}
(59)	(59)
(60)	(60)

5. 第1図の入力データを読み、次の処理を行う。

- 農作物コードをもとに配列 Ncd を探索し、配列 Syu に収穫量計を求める。なお、Syu の48列目には合計を求める。また、Syu の行方向の添字は Ncd の添字と対応し、列方向の添字は都道府県コードと対応している。

配列

Syu	(0)	(1)	(2)	～	(46)	(47)	(48)
(0)				～			
(1)				～			
(2)				～			
}	}	}	}	}	}	}	}
(59)				～			
(60)				～			

(合計)

- 配列 Bsyu に分類ごとに収穫量計を求める。なお、Bsyu の添字は分類番号と対応している。

配列

Bsyu	(0)	(1)	(2)	(3)	(4)

6. 入力データが終了したら、次の処理を行う。

- 分類が変わるごとに分類別収穫量計を第2図のように表示する。
- 配列 Work を利用して、農作物ごとに都道府県別収穫量計の降順に並べ替える。

配列

Work	(0)	(1)	(2)	～	(46)	(47)
				～		

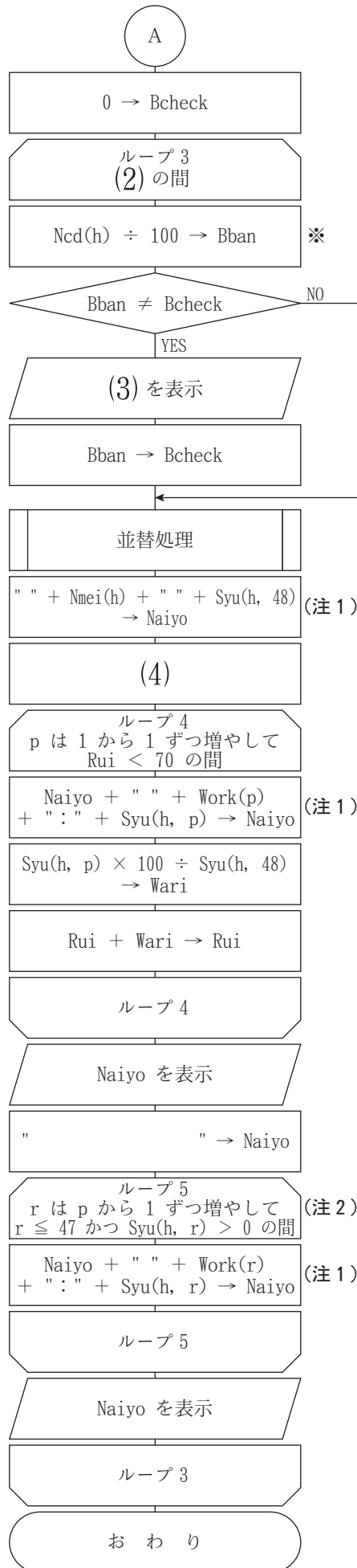
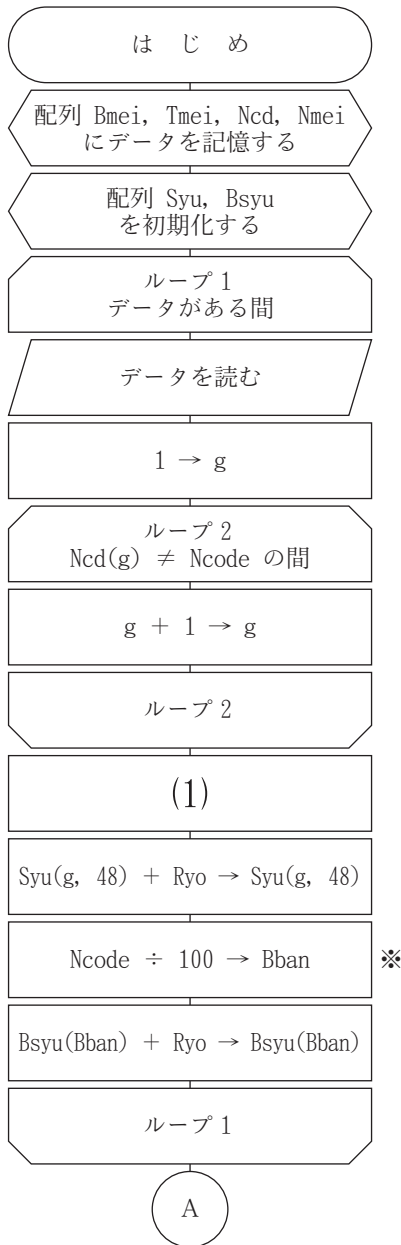
- 農作物別収穫量計と収穫量上位の都道府県：収穫量計を第2図のように表示する。なお、収穫量上位の都道府県：収穫量計は、農作物別収穫量計に対する割合とその累計を求め、累計が70%以下のものについて収穫量計の降順に表示する。
- その他の都道府県：収穫量計を第2図のように収穫量計の降順に表示する。なお、収穫量計が 0 の場合、表示しない。

7. データにエラーはないものとする。

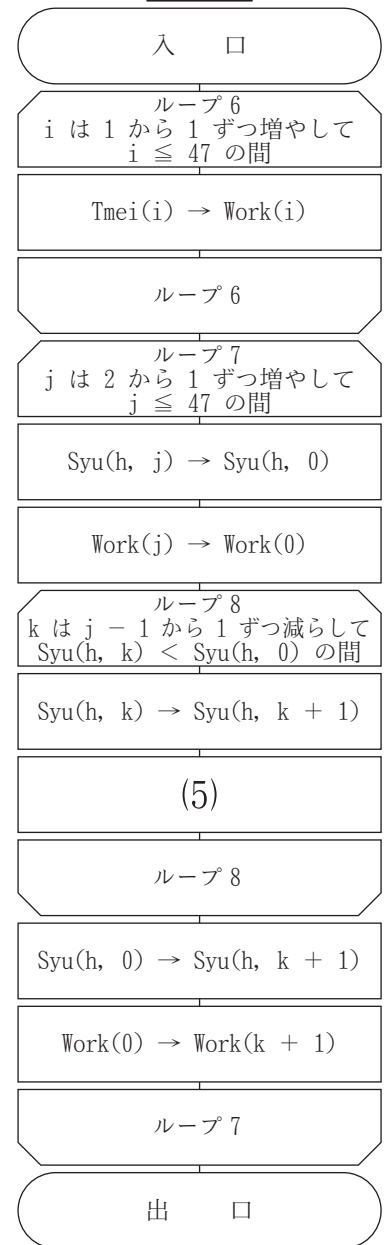
解答群

- | | |
|----------------------------------|--|
| ア. Bmei(Bban), Bsyu(Bban) | イ. Bmei(Bcheck), Bsyu(Bcheck) |
| ウ. 0 → Wari | エ. h は 1 から 1 ずつ増やして $h \leq 60$ |
| オ. Work(k) → Work(k + 1) | カ. Syu(g, Ncode) + Ryo → Syu(g, Ncode) |
| キ. h は 1 から 1 ずつ増やして $h \leq 47$ | ク. 0 → Rui |
| ケ. Work(k) → Work(k - 1) | コ. Syu(g, Tcode) + Ryo → Syu(g, Tcode) |

< 流れ図 >



並替処理



※ 小数点以下切り捨て

(注1) ここでの「+」は、文字列結合を意味する。

(注2) 条件式が「かつ」で複合されている場合、先に記述された条件式が偽になった時点で、判定を終了する。

【7】 プログラムの説明を読んで、プログラムの(1)～(5)を答えなさい。

<プログラムの説明>

処理内容

ある地域の道の駅の1か月分の売上データを読み、集計結果を表示する。

入力データ

道の駅データ（ファイル名：roadsidestation.csv）

道の駅コード	道の駅名	売場面積 (㎡)
×××	×～×	××××.×

(第1図)

売上データ（ファイル名：selling.csv）

月日	道の駅コード	伝票番号	売上金額
××××	×××	×××××	×～×

(第2図)

実行結果

道の駅番号(1～10)を入力してください。→3
 きらきら高原の森【電気自動車充電スタンドあり】（売場面積 (㎡) : 1,160.5)
 売上金額合計: 61,227,532 (2位) 売場効率: 52,759.6 (6位)
 売場区分名 売上金額計 構成比 (%)
 軽食 8,510,627 13.9
 レストラン 11,388,321 18.6
 地域特産品 8,755,537 14.3
 菓子・飲料 10,776,045 17.6
 その他土産物 21,797,002 35.6
 電気自動車充電売上金額計: 199,260 (369件)
 道の駅番号(1～10)を入力してください。→4
 ゆめの里【電気自動車充電スタンドなし】（売場面積 (㎡) : 1,217.2)
 }

(第3図)

処理条件

- 第1図の道の駅データは、道の駅コードの昇順に記録されている。なお、道の駅コードは次の例のように構成されており、道の駅番号は1～10の10種類、電気自動車充電スタンド区分は0（電気自動車充電スタンドなし）、1（電気自動車充電スタンドあり）である。

例 101 → 1 0 1
 道の駅番号 電気自動車充電スタンド区分

- 第2図の売上データは、月日、道の駅コード、伝票番号の昇順に記録されている。なお、伝票番号は次の例のように構成されており、売場区分は1（軽食）～5（その他土産物）および6（電気自動車充電）である。

例 40215 → 4 0 2 1 5
 売場区分 連番

- 第1図の道の駅データを読み、道の駅情報を管理するRoadsideStationクラスをインスタンス化し、配列objに記憶する。なお、objの添字は道の駅番号と対応している。

配列

obj	(0)	(1)	(2)	～	(9)	(10)
				～		

- 配列sFloorに売場区分名を記憶する。なお、sFloorの添字は売場区分と対応している。

配列

sFloor	(0)	(1)	(2)	(3)	(4)	(5)
		軽食	レストラン	地域特産品	菓子・飲料	その他土産物

- 第2図の売上データを読み、売場区分が1～5の場合、配列totalに売上金額計を求め、total(0)には売上金額合計を求める。なお、totalの添字は売場区分と対応している。また、売場区分が6の場合、電気自動車充電の売上金額計と件数を求める。

配列

total	(0)	(1)	(2)	(3)	(4)	(5)

(合計)

- データを読み終えたあと、次の処理を行う。

- 売場効率を次の計算式で求める。

売場効率 = 売上金額合計 ÷ 売場面積 (㎡)

- 売上金額合計の降順に道の駅間の順位をつける。なお、売上金額合計が同じ場合は同順位とする。また、売場効率についても同様に処理を行う。

- キーボードから道の駅番号が入力されたら、売場区分ごとの構成比(%)を次の計算式で求め、第3図のようにディスプレイに表示する。

構成比(%) = 売上金額計 × 100 ÷ 売上金額合計

- キーボードから0が入力されたら処理を終了する。

< J a v a プログラム >

```

//クラスRoadsideStation
public class RoadsideStation {
    private static String[] sFloor = { "", "軽食", "レストラン", "地域特産品", "菓子・飲料", "その他土産物" };
    private int[] total = new int[6];
    private String rsName;
    private double rsArea, efficiency;
    private int evCharge, sRank, eRank, evChargeTotal, evChargeCount;
    public RoadsideStation(String rsName, double rsArea, int evCharge) {
        this.rsName = rsName; this.rsArea = rsArea; this.evCharge = evCharge;
        if(evCharge == 0) {
            this.rsName = this.rsName + "【電気自動車充電スタンドなし】";
        } else {
            this.rsName = this.rsName + "【電気自動車充電スタンドあり】";
        }
        sRank = 1; eRank = 1;
    }
    public int getTotal() {
        return total[0];
    }
    public double getEfficiency() {
        return efficiency;
    }
    public void sRanking() {
        sRank += 1;
    }
    public void eRanking() {
        eRank += 1;
    }
    public void calcTotal(int floorCode, int sales) {
        total[floorCode] += sales;
        (1);
    }
    public void calcTotal(int sales) {
        evChargeTotal += sales;
        evChargeCount += 1;
    }
    public void calcEfficiency() {
        efficiency = total[0] / rsArea;
    }
    public void printResult() {
        System.out.printf("%s (売場面積 (㎡) : %.7lf) ¥n", rsName, rsArea);
        System.out.printf("売上金額合計 : %.11d (%2d位) 売場効率 : %.9lf (%2d位) ¥n", total[0], sRank, efficiency, eRank);
        System.out.println(" 売場区分名 売上金額計 構成比 (%) ");
        for(int i = 1; i <= 5; i++) {
            double ratio = (double) total[i] * 100 / total[0];
            System.out.printf(" %6s %10d %4.1f¥n", (2), total[i], ratio);
        }
        if(evCharge == 1) {
            System.out.printf("電気自動車充電売上金額計 : %.9d (%.5d件) ¥n", evChargeTotal, evChargeCount);
        }
    }
}

//クラスSalesAnalysis
import java.io.BufferedReader;

public class SalesAnalysis {
    private static RoadsideStation[] obj = new RoadsideStation[11];
    public static void main(String[] args) {
        try {
            BufferedReader fileIn1 = new BufferedReader(new FileReader("roadsidestation.csv"));
            String line;
            while((line = fileIn1.readLine()) != null) {
                String[] str = line.split(",");
                int rsCode = Integer.parseInt(str[0]);
                String rsName = str[1];
                double rsArea = Double.parseDouble(str[2]);
                int rsNumber = rsCode / 10;
                int evCharge = rsCode % 10;
                obj[rsNumber] = new RoadsideStation(rsName, rsArea, evCharge);
            }
            fileIn1.close();
            BufferedReader fileIn2 = new BufferedReader(new FileReader("selling.csv"));
            while((line = fileIn2.readLine()) != null) {
                String[] str = line.split(",");
                int rsCode = Integer.parseInt(str[1]);
                int slipNumber = Integer.parseInt(str[2]);
                int sales = Integer.parseInt(str[3]);
                int rsNumber = rsCode / 10;
                int floorCode = slipNumber / 10000;
                if(floorCode <= 5) {
                    (3);
                } else {
                    obj[rsNumber].calcTotal(sales);
                }
            }
            fileIn2.close();
        } catch(Exception e) {
            System.out.println("エラーが発生しました" + e);
        }
        for(int g = 1; (4); g++) {
            obj[g].calcEfficiency();
        }
        for(int h = 1; h <= 9; h++) {
            for(int j = h + 1; j <= 10; j++) {
                if((5)) {
                    obj[h].sRanking();
                } else if(( 解答不要 )) {
                    obj[j].sRanking();
                }
            }
            (注) 省略されている処理の詳細は処理条件 6
        }
    }
}

Scanner keyboardIn = new Scanner(System.in);
System.out.print("道の駅番号 (1~10) を入力してください。→"); int rsNumber = keyboardIn.nextInt();
while(rsNumber != 0) {
    obj[rsNumber].printResult();
    System.out.print("道の駅番号 (1~10) を入力してください。→"); rsNumber = keyboardIn.nextInt();
}
keyboardIn.close();
}

```

【7】 プログラムの説明を読んで、プログラムの(1)～(5)を答えなさい。

<プログラムの説明>

処理内容

ある通訳者派遣会社の1か月分の営業データを読み、通訳区分別集計結果と通訳者別一覧、言語別一覧を表示する。

入力データ

通訳者データ (ファイル名: tuyaku.csv)

通訳者番号	通訳者名	言語区分
×××	×～×	××

(第1図)

営業データ (ファイル名: eigyo.csv)

日時	通訳者番号	通訳区分	通訳時間
×～×	×××	×	××

(第2図)

通訳区分別集計結果

☐同時通訳 (OptionButton1)
 ☐逐次通訳 (OptionButton2)
 ☒ささやき通訳 (OptionButton3)
 ☐観光通訳 (OptionButton4)
 ☐合計 (OptionButton5)

40,560,000	34,122,000	12,416,000	10,598,000	97,696,000
------------	------------	------------	------------	------------

(Label1)

表示 終了

通訳者別一覧

順位	通訳者名	料金計
1位	原田 知	360,000
2位	大槻 ひとみ	336,000
3位	塩田 裕康	320,000
⋮	⋮	⋮
95位	岡野 雄	40,000
⋮	⋮	⋮

(TextBox1)

言語別一覧

言語名	料金計	比率(%)
英語	7,208,000	58.1
フランス語	2,040,000	16.4
ドイツ語	528,000	4.3
⋮	⋮	⋮
ポルトガル語	616,000	5.0

(TextBox2)

(第3図)

処理条件

- 第1図の通訳者データは通訳者番号の昇順に記録されている。なお、通訳者番号は1からの連番であり、通訳者は150人以下である。また、言語区分は1(英語)～10(ポルトガル語)である。
- 第2図の通訳区分は1(同時通訳)～4(観光通訳)である。
- ユーザーフォーム初期化時に、次の処理を行う。
 - 配列 Gmei に言語名を、配列 Kryokin に基本料金を記憶する。なお、基本料金は通訳時間が8時間までの料金である。また、Gmei の添字は言語区分と、Kryokin の添字は通訳区分と対応している。

配列

	Gmei	Kryokin	(0)	(1)	(2)	(3)	(4)
(0)							
(1)	英語			75000	55000	40000	35000
⋮	⋮			(同時通訳)	(逐次通訳)	(ささやき通訳)	(観光通訳)
(10)	ポルトガル語						

- 第1図の通訳者データを読み、配列 Tmei に通訳者名を、配列 Gkubun に言語区分を記憶する。なお、Tmei、Gkubun の添字は通訳者番号と対応している。

配列

	Tmei	Gkubun
(0)		(0)
(1)	橋 アラン	(1)
⋮	⋮	⋮
(149)	田中 さき	(149)
(150)		(150)

- 第2図の営業データを読み、次の処理を行う。
 - 基本料金から料金を求める。なお、通訳時間が8時間を超えた場合、次の計算式で、延長時間1時間あたり基本料金の20%を追加する。

$$\text{料金} = \text{基本料金} + \text{延長時間} \times (\text{基本料金} \times 0.2)$$
 - 配列 Tkei に通訳者ごとに、配列 Gkei に言語ごとに料金計を求める。なお、Tkei の0列目、Gkei の0行目と0列目には合計を求める。また、第3図の通訳区分別集計結果は、同時通訳 (OptionButton1)～観光通訳 (OptionButton4) および合計 (OptionButton5) であり、Tkei の行方向の添字は通訳者番号と、Gkei の行方向の添字は言語区分と対応し、Tkei、Gkei の列方向の添字は通訳区分と対応している。

配列

	(0)	(1)	～	(4)
Tkei			～	
(0)			～	
(1)			～	
⋮	⋮	⋮	⋮	⋮
(149)			～	
(150)			～	
	(合計)	(同時通訳)	～	(観光通訳)

	(0)	(1)	～	(4)
Gkei			～	
(0)			～	(合計)
(1)			～	
⋮	⋮	⋮	⋮	⋮
(10)			～	
	(合計)	(同時通訳)	～	(観光通訳)

- データを読み終えたあと、配列 Jun を利用して通訳区分ごとに料金計の降順に順位をつける。なお、料金計が同じ場合は同順位とする。また、Jun と配列 Tkei の添字は対応している。

配列

	(0)	(1)	～	(4)
Jun			～	
(0)			～	
(1)			～	
⋮	⋮	⋮	⋮	⋮
(149)			～	
(150)			～	
	(合計)	(同時通訳)	～	(観光通訳)

- 通訳区分別集計結果を Label1 に表示する。
- 第3図のように通訳区分を指定し、「表示」ボタンをクリックすると、次の処理を行う。
 - TextBox1 に順位順に通訳者別一覧を表示する。なお、料金が0の場合、表示しない。
 - 言語別料金計の比率(%)を次の計算式で求め、TextBox2 に言語別一覧を表示する。

$$\text{言語別料金計の比率(\%)} = \text{言語別料金計} \times 100 \div \text{言語別料金計の合計}$$

< マクロ言語プログラム >

```
Option Explicit
Dim Gmei(10) As String, Tkb As Long, Tmei(150) As String, Nin As Long, ~ Gkei(10, 4) As Double
```

```
Private Sub UserForm_Initialize()
    Dim Kryokin(4) As Long, Tuban As Long, Tumei As String, Gekubun As Long, Gkubun(150) As Long, h As Long, ~ n As Long
    Gmei(1) = "英語"           ": Gmei(2) = "フランス語"   ": ~ Gmei(10) = "ポルトガル語"
    Kryokin(1) = 75000: Kryokin(2) = 55000: Kryokin(3) = 40000: Kryokin(4) = 35000
    OptionButton5.Value = True: Tkb = 0
    Label1.Caption = "": TextBox1.Text = "": TextBox2.Text = ""
    Open ThisWorkbook.Path & "¥tuyaku.csv" For Input As #1
    Do While Not EOF(1)
        Input #1, Tuban, Tumei, Gekubun
        Tmei(Tuban) = Tumei
        Gkubun(Tuban) = Gekubun
    Loop
    Nin = Tuban
    Close #1
    For h = 1 To Nin
        For j = 0 To 4
            Jun(h, j) = 1
        Next j
    Next h
    Open ThisWorkbook.Path & "¥eigyo.csv" For Input As #2
    Do While Not EOF(2)
        Input #2, Nitiji, Tuban, Tukubun, Tujikan
        Ryokin = Kryokin(Tukubun)
        If Tujikan > 8 Then
            Ryokin = Ryokin + (Tujikan - 8) * (Ryokin * 0.2)
        End If
        Tkei(Tuban, Tukubun) = Tkei(Tuban, Tukubun) + Ryokin
        (1)
        Gkei(Gkubun(Tuban), Tukubun) = Gkei(Gkubun(Tuban), Tukubun) + Ryokin
        Gkei(Gkubun(Tuban), 0) = Gkei(Gkubun(Tuban), 0) + Ryokin
        Gkei(0, Tukubun) = Gkei(0, Tukubun) + Ryokin
        Gkei(0, 0) = Gkei(0, 0) + Ryokin
    Loop
    Close #2
    For j = 0 To 4
        For m = 1 To Nin - 1
            For n = m + 1 To Nin
                If (2) Then
                    Jun(m, j) = Jun(m, j) + 1
                ElseIf (3) Then
                    Jun(n, j) = Jun(n, j) + 1
                End If
            Next n
        Next m
    Next j
    For j = 1 To 4
        Label1.Caption = Label1.Caption & " " & Format(Format((4), "##,###,##0"), "#####") & " "
    Next j
    Label1.Caption = Label1.Caption & " " & Format(Format(Gkei(0, 0), "###,###,##0"), "#####")
End Sub
```

```
Private Sub 表示_Click()
    Dim p As Long, r As Long, s As Long, Hiritu As Double
    TextBox1.Text = "": TextBox2.Text = ""
    For p = 1 To Nin
        For r = 1 To Nin
            If (4) = p And Tkei(r, Tkb) > 0 Then
                TextBox1.Text = TextBox1.Text & Format(Format(p, "##0"), "@@") & "位 " & Tmei(r) & " " &
                    Format(Format(Tkei(r, Tkb), "###,##0"), "#####") & Chr(13) & Chr(10)
            End If
        Next r
    Next p
    For (5)
        Hiritu = Gkei(s, Tkb) * 100 / Gkei(0, Tkb)
        TextBox2.Text = TextBox2.Text & Gmei(s) & " " & Format(Format(Gkei(s, Tkb), "##,###,##0"), "#####") & " " &
            Format(Format(Hiritu, "##0.0"), "@@@") & Chr(13) & Chr(10)
    Next s
End Sub
```

```
Private Sub 終了_Click()
    End
End Sub
```

```
Private Sub OptionButton1_Click()
    Tkb = 1
End Sub
```

```
}
```

```
Private Sub OptionButton4_Click()
    Tkb = 4
End Sub
```

```
Private Sub OptionButton5_Click()
    Tkb = 0
End Sub
```

(平成31年 1 月20日実施)

主催 公益財団法人 全国商業高等学校協会

平成30年度（第60回）情報処理検定試験プログラミング部門 第1級
解 答 用 紙

【1】

1	2	3	4	5

【2】

1	2	3	4	5

【3】

1	2	3	4	5

小 計

【4】

(1)	(2)	(3)	(4)	(5)

【5】

(1)	(2)	(3)	(4)	(5)

【6】

(1)	(2)	(3)	(4)	(5)

小 計

..... [J a v a] ・ [マ ク ロ 言 語]

【7】

(1)	
(2)	
(3)	
(4)	
(5)	

試験場校名	受 験 番 号	選 択 言 語		小 計	合 計
		J a v a	マ ク ロ 言 語		

選択言語を で囲むこと

(平成31年 1 月20日実施)

主催 公益財団法人 全国商業高等学校協会

平成30年度（第60回）情報処理検定試験プログラミング部門 第1級

審 査 基 準

【1】

1	2	3	4	5
シ	サ	エ	ウ	ク

各2点
計10点

【2】

1	2	3	4	5
キ	コ	ウ	オ	ケ

各2点
計10点

【3】

1	2	3	4	5
ウ	イ	ア	ア	イ

各2点
計10点

小 計
30

【4】

(1)	(2)	(3)	(4)	(5)
力	ウ	イ	エ	ア

各3点
計15点

【5】

(1)	(2)	(3)	(4)	(5)
ケ	ア	キ	イ	コ

各3点
計15点

【6】

(1)	(2)	(3)	(4)	(5)
コ	エ	ア	ク	オ

各3点
計15点

小 計
45

..... [J a v a] ・ [マ ク ロ 言 語]

[J a v a] (注) =, 演算子の前後の空白は問わない。

【7】

(1)	t o t a l [0] += s a l e s
(2)	s F l o o r [i]
(3)	o b j [r s N u m b e r] . c a l c T o t a l (f l o o r C o d e , s a l e s)
(4)	g <= 1 0
(5)	o b j [h] . g e t T o t a l () < o b j [j] . g e t T o t a l ()

[マ ク ロ 言 語] (注) 大文字, 小文字および=, 演算子の前後の空白は問わない。

【7】

(1)	T k e i (T u b a n , 0) = T k e i (T u b a n , 0) + R y o k i n
(2)	T k e i (m , j) < T k e i (n , j)
(3)	G k e i (0 , j)
(4)	J u n (r , T k b)
(5)	s = 1 T o 1 0

各5点 計25点

小 計
25

合 計
100