

2019年 9 月22日 実施

令和元年度（第61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

受験番号

--

【 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. 試作品を作成し、ユーザからの評価を受け、改良することにより、システムの開発を行う手法。
2. コンピュータで数値を扱う際、四捨五入などの端数処理を行った結果が、本来の計算結果とわずかに異なる現象。
3. インターネットで標準的に利用され、OSなどの環境が異なるコンピュータ間でも通信を可能とするプロトコル。
4. ブラウザを通じて、Webサイトの閲覧日時やアクセス回数などユーザの識別に使われる情報が、クライアント側の端末に一時的に保存されるしくみ。
5. システムの評価指標に使われる、コンピュータに処理の要求をしてからすべての結果が出力されるまでの時間。

解答群

- | | | |
|----------------|-------------|--------------|
| ア. スループット | イ. キュー | ウ. プロトタイプモデル |
| エ. ウォータフォールモデル | オ. 情報落ち | カ. Cookie |
| キ. ターンアラウンドタイム | ク. レスポンスタイム | ケ. RASIS |
| コ. TCP/IP | サ. 丸め誤差 | シ. クロック周波数 |

【 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. フェールセーフ 2. OR回路 3. 機能テスト
 4. プログラム設計 5. ポインタ

< B 群 >

- ア. プログラミングにおいて、目的のデータが格納されたアドレスや添字を示すもの。
- イ. 上位のモジュールから、順次結合しながら検証を行うテスト。
- ウ. システム開発において、利用者が要求する項目について調査・分析・整理したものをもとに、入出力画面や帳票を設計する工程。
- エ. ユーザから要求された仕様を、コンピュータシステムが満たしているかどうかを検証するテスト。
- オ. 後に格納されたデータが、先に取り出されるデータ構造。
- カ. 故障や障害が生じないように、個々の構成要素の品質を高め、故障などの原因となる要素をできるだけ排除することで信頼性を高める考え方。
- キ. 二つの入力がかともに「1」のときのみ、「1」を出力する論理回路。
- ク. 安全確保を優先し、故障時や異常発生時にシステムを停止させるなど、被害を最小限にする考え方。
- ケ. システム開発において、内部設計で作成された仕様書や処理の手順に基づき構造設計を行う工程。
- コ. 二つの入力のうち少なくとも一方が「1」のとき、「1」を出力する論理回路。

【 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. 16進数の0.7を10進数で表したもの。
 ア. 0.4375 イ. 0.5625 ウ. 0.875
2. コンピュータシステムが1秒間に処理できる命令を、100万単位で表した性能指標。
 ア. NAS イ. UPS ウ. MIPS
3. 複数の磁気ディスク装置などを1つの装置として管理し、アクセスの高速化や、信頼性の向上を図る技術。
 ア. RAID イ. DMZ ウ. DHCP
4. 公開鍵暗号方式などを利用し、通信したデータの改ざんやなりすましが無いことを確認するしくみ。
 ア. VoIP イ. デジタル署名 ウ. プラグアンドプレイ
5. 0.75Gbpsの通信回線を使用して、12GBのデータを転送するのに必要な時間は何秒か。ただし、1GB = 10⁹Bとする。なお、伝送効率は50%とし、その他の外部要因は考えないものとする。
 ア. 128秒 イ. 256秒 ウ. 512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に順位をつけてディスプレイに表示する。

処理条件

1. 配列 Atai に数値を記憶する。なお、データ件数は n 件である。

配列

Atai	(0)	(1)	~	(n-2)	(n-1)
	36.7	22.8	~	19.1	40.6

2. 配列 Jun を利用し、配列 Atai の数値の昇順に順位をつける。なお、数値が同じ場合は同順位とする。

配列

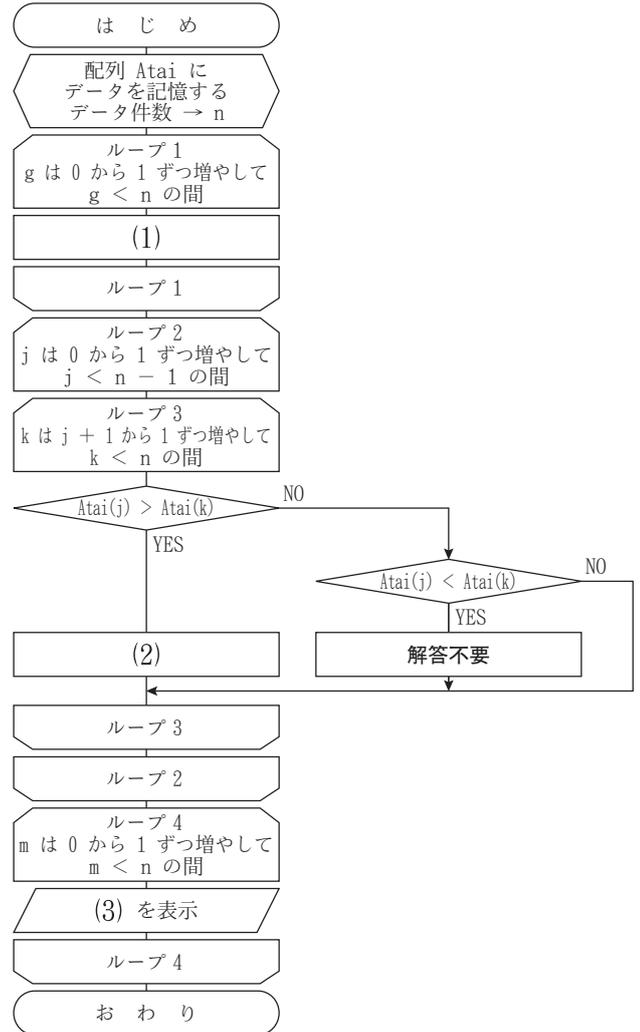
Jun	(0)	(1)	~	(n-2)	(n-1)
			~		

3. 順位をつけ終わったら、配列 Atai と Jun の内容を表示する。

解答群

- ア. Atai(n), Jun(n)
- イ. $1 \rightarrow \text{Jun}(g)$
- ウ. $0 \rightarrow \text{Jun}(g)$
- エ. $\text{Jun}(k) + 1 \rightarrow \text{Jun}(k)$
- オ. Atai(m), Jun(m)
- カ. $\text{Jun}(j) + 1 \rightarrow \text{Jun}(j)$

<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 Kazu に数値を記憶する。なお、データ件数は n 件である。

配列

Kazu	(0)	(1)	~	(n-2)	(n-1)
	285	458	~	483	326

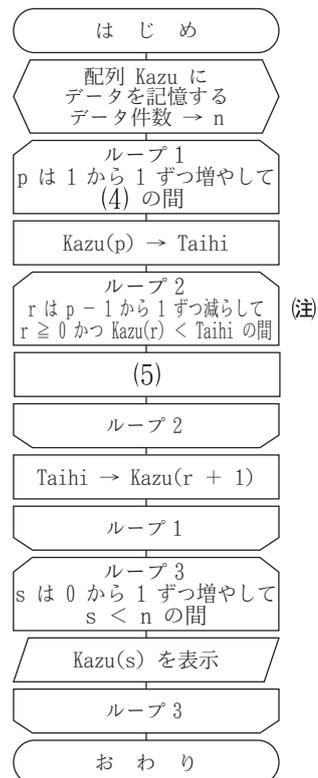
2. 配列 Kazu の数値を降順に並べ替える。
3. 並べ替えが終わったら、配列 Kazu の内容を表示する。

解答群

- ア. $p < n - 1$
- イ. $\text{Kazu}(r + 1) \rightarrow \text{Kazu}(r)$
- ウ. $\text{Kazu}(r) \rightarrow \text{Kazu}(r + 1)$
- エ. $p < n$

(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>



【 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

あるイベントスペースの1年間の来場者データを読み、集計結果をディスプレイに表示する。

入力データ

分類番号 (Bban)	イベント番号 (Iban)	来場日 (Rhi)
×	××	××××

(第1図)

実行結果

(イベント来場者集計結果)					
(分類名)	(イベント名)	(開始日)	(日数)	(来場者数)	(平均来場者数)
美術	北欧イラスト展	0103	8	1,652	206
	}	}	}	}	}
	ひまわり絵画祭	1203	9	2,124	236
(分類別来場者数)	25,627	(分類別平均来場者数)	210		
音楽	春のクラシック	0116	5	981	196
	}	}	}	}	}
	JPOPフェア	1123	10	1,590	159
(分類別来場者数)	10,227	(分類別平均来場者数)	189		

(第2図)

処理条件

- 第1図のデータは分類番号、イベント番号、来場日の昇順に記録されている。なお、分類番号は1～4、イベント番号は1～55である。また、来場日は次の例のように構成されており、各イベントの開始日から最終日まですべてで来場があり、同一日に複数のイベントはないものとする。

例 0501 → 5月1日

- 配列 Bmei に分類名を、配列 Imei にイベント名を記憶する。なお、Bmei の添字は分類番号と対応し、Imei の添字はイベント番号と対応している。

配列

Bmei	(0)	(1)	～	(4)
		美術	～	音楽
Imei	(0)	(1)	～	(55)
		北欧イラスト展	～	落語寄席

- 第1図の入力データを読み、次の処理を行う。
 - 分類番号が変わるごとに分類名を表示する。
 - イベントごとの来場者数を集計する。
 - 来場日が変わるごとに日数を集計する。
 - イベント番号が変わるごとに分類別来場者数と分類別日数の計を求める。なお、平均来場者数を次の計算式で求め、イベント名から平均来場者数を第2図のように表示する。

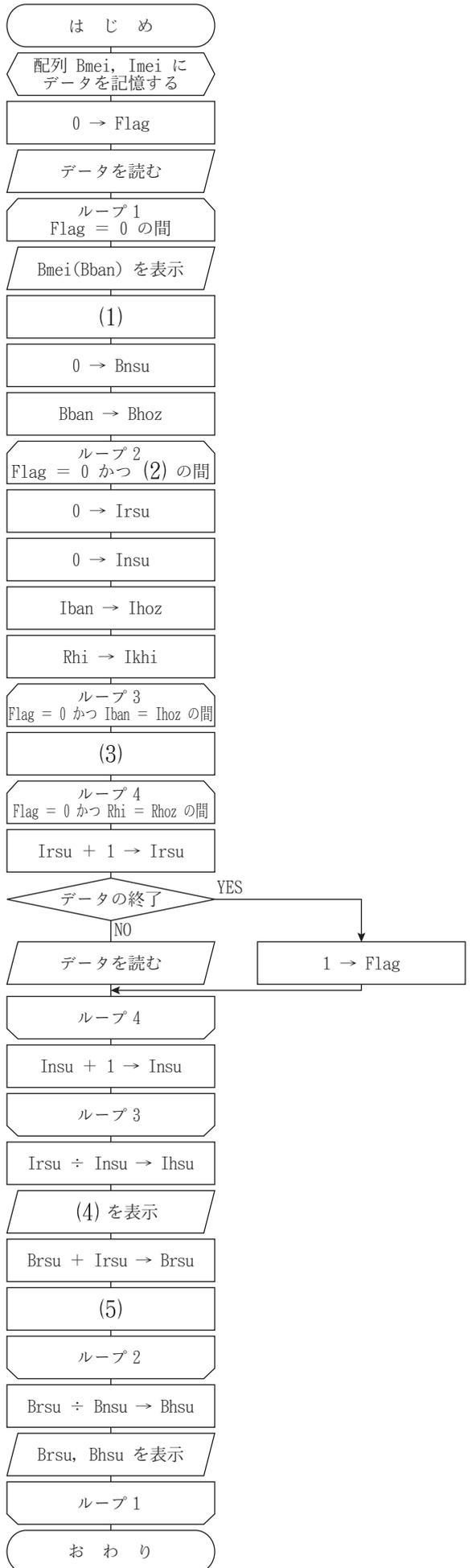
$$\text{平均来場者数} = \text{来場者数} \div \text{日数}$$
 - 分類番号が変わるごとに分類別平均来場者数を次の計算式で求め、分類別来場者数とともに、第2図のように表示する。

$$\text{分類別平均来場者数} = \text{分類別来場者数} \div \text{分類別日数の計}$$
- データにエラーはないものとする。

解答群

- ア. Bban → Brsu
- イ. Bban = Bhoz
- ウ. Imei(Bhoz), Ikhi, Insu, Irsu, Ihsu
- エ. 0 → Brsu
- オ. Bnsu + Insu → Bnsu
- カ. Ikhi → Rhoz
- キ. Bnsu + 1 → Bnsu
- ク. Rhi → Rhoz
- ケ. Imei(Ihoz), Ikhi, Insu, Irsu, Ihsu
- コ. Iban = Bhoz

<流れ図>



【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

貸衣装店の1か月分の貸出データを読み、集計結果をディスプレイに表示する。

入力データ

日付 (Hiduke) ××××	店コード (Miseco) ×	貸衣装コード (Kasico) ××××
------------------------	-----------------------	----------------------------

(第1図)

実行結果

(貸出回数一覧表) (貸衣装名)	(合計)	(Web店)	(東店)	(西店)	(南店)	(北店)
フォーマルA黒	11	6	1	0	2	2
フォーマルA白	22	14	3	3	2	0
}	}	}	}	}	}	}
キッズG青	9	2	3	1	3	0

(貸衣装名)
貸出なし フォーマルF赤 セミフォーマルA白 ~ キッズA桃

(各店の人気貸衣装)
(店コードをキーボードから入力) 1
(店名) Web店
(貸出回数) (貸衣装名)
15 フォーマルD黒
14 フォーマルA白 カジュアルA黒
12 セミフォーマルB黒 カジュアルD桃 ~ キッズC青
(店コードをキーボードから入力) 3
(店名) 西店
(貸出回数) (貸衣装名)
6 セミフォーマルB黒
5 カジュアルA黒 キッズA黒
3 フォーマルA白 カジュアルD桃
(店コードをキーボードから入力) 0

(第2図)

処理条件

- 第1図の店コードは1(Web店)～5(北店)である。なお、すべての店のデータがある。
- 配列Mmに店名を記憶する。なお、Mmの添字は店コードと対応している。

配列

Mm	(0)	(1)	(2)	(3)	(4)	(5)
		Web店	東店	西店	南店	北店

- 配列Kcに貸衣装コードを、配列Kmに貸衣装名を、貸衣装コードの昇順に記憶する。なお、貸衣装は100種類以下であり、KcとKmの添字は対応している。

配列

Kc		Km	
(0)		(0)	
(1)	1011	(1)	フォーマルA黒
}	}	}	}
(50)	3027	(50)	カジュアルD桃
}	}	}	}
(100)		(100)	

- 第1図の入力データを読み、次の処理を行う。

- 貸衣装コードをもとに配列Kcを探索し、店別貸衣装別に配列Kkに貸出回数を求める。なお、Kkの0列目には合計を求める。また、Kkの行方向の添字はKcの添字と対応し、列方向の添字は店コードと対応している。

配列

Kk	(0)	(1)	(2)	(3)	(4)	(5)
(0)						
(1)						
}	}	}	}	}	}	}
(50)						
}	}	}	}	}	}	}
(100)						

(合計)

- 入力データが終了したら、次の処理を行う。

- 貸衣装名と各店の貸出回数を、第2図のように表示する。なお、貸出回数の合計が0の場合、表示しない。
- 貸出回数の合計が0の貸衣装名を、第2図のようにすべて表示する。
- 店コードをキーボードから入力すると、最大の貸出回数から3番目の貸出回数までを配列Mxkkに、貸衣装名を配列Mxkmに求め、貸出回数と貸衣装名を、第2図のように表示する。なお、同じ貸出回数がある場合は、すべての貸衣装名を表示する。また、MxkkとMxkmの添字は対応している。

配列

Mxkk	(0)	(1)	(2)	(3)
Mxkm	(0)	(1)	(2)	(3)

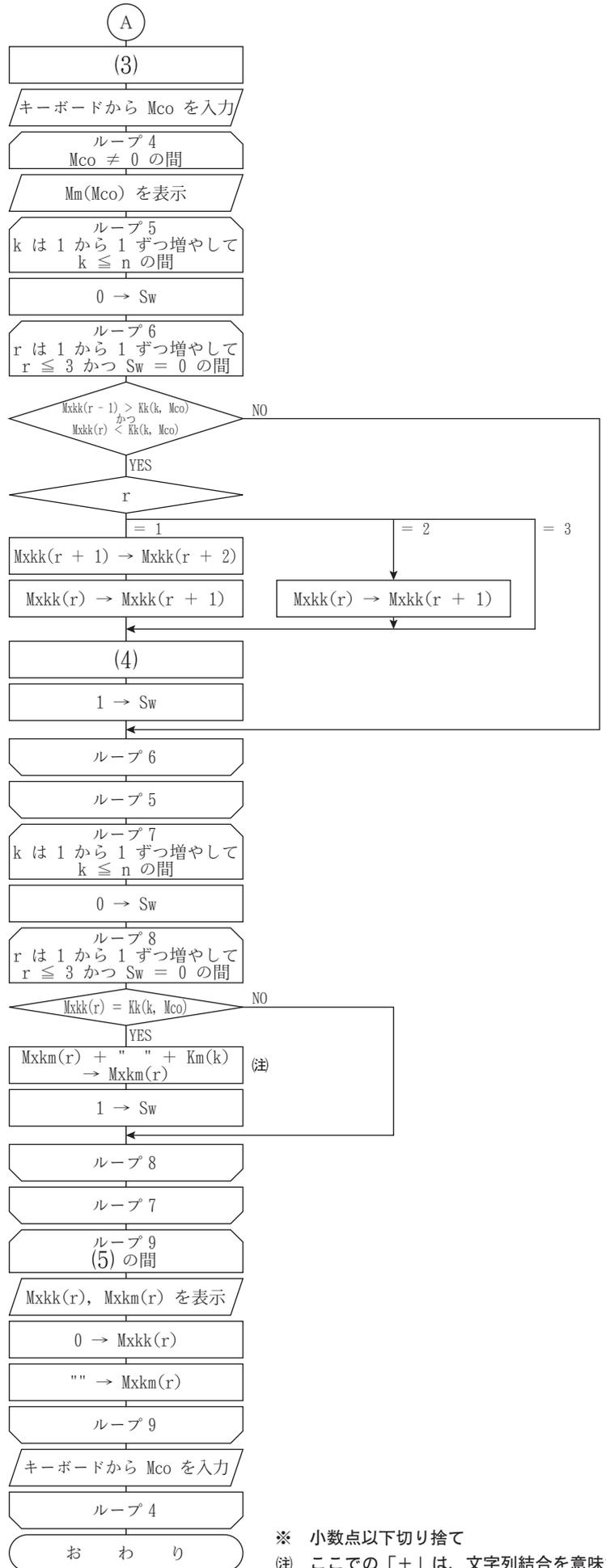
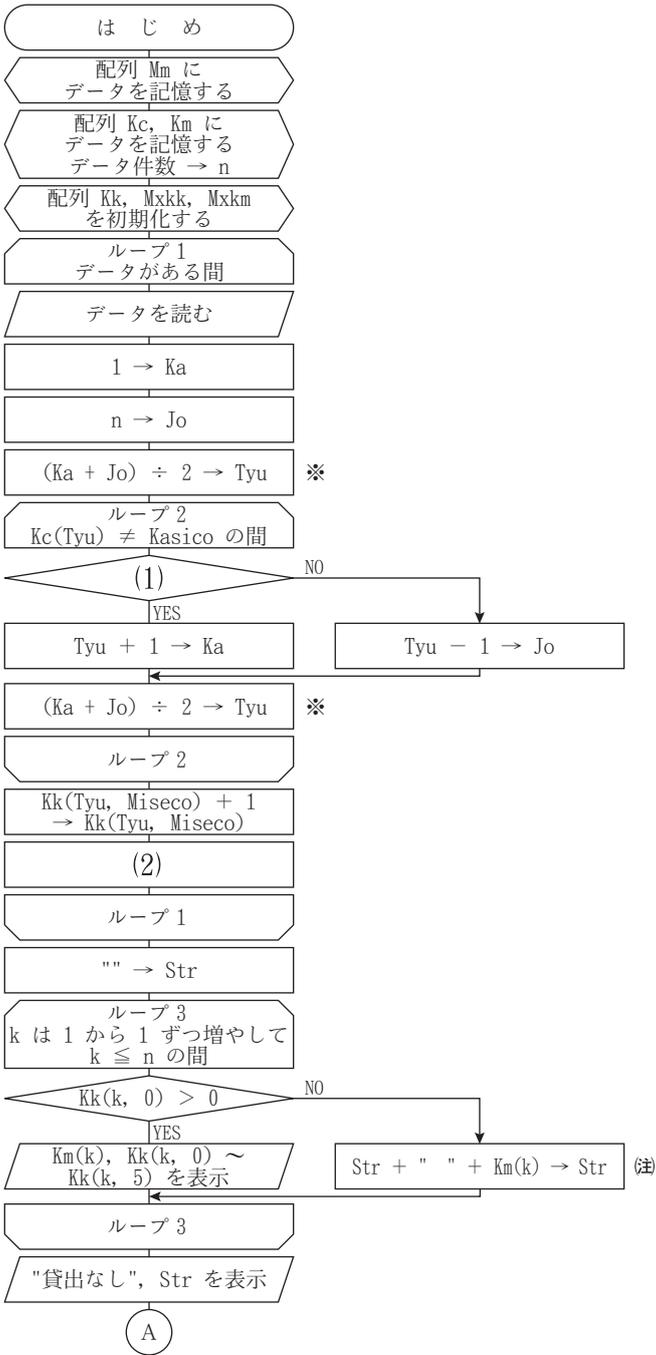
- キーボードから店コードとして0が入力されたら処理を終了する。

- データにエラーはないものとする。

解答群

- | | |
|---------------------------------|--|
| ア. Kk(k, Mco) → Mxkk(r) | イ. 0 → Mxkk(0) |
| ウ. 999 → Mxkk(0) | エ. Kk(Tyu, 0) + 1 → Kk(Tyu, 0) |
| オ. r は 0 から 1 ずつ増やして $r \leq 3$ | カ. Kc(Tyu) > Kasico |
| キ. Kc(Tyu) < Kasico | ク. Kk(Tyu, Kasico) + 1 → Kk(Tyu, Kasico) |
| ケ. Kk(k, r) → Mxkk(r) | コ. r は 1 から 1 ずつ増やして $r \leq 3$ |

< 流れ図 >



※ 小数点以下切り捨て
 (注) ここでの「+」は、文字列結合を意味する。

【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

ある企業向け弁当販売店の契約先データと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

契約先データ (client.csv)

契約先コード ××××	契約先名 ×～×
----------------	-------------

(第1図)

売上データ (selling.csv)

日付 ××××	契約先コード ××××	商品番号 ××	数量 ××
------------	----------------	------------	----------

(第2図)

実行結果

店舗番号(1~9)を入力してください→2				
佐野店 【売上金額合計：1,499,400】				
(契約先名)	(数量計)	(売上金額計)	(売上比率)	(累積売上比率)
〇〇株式会社	209	97,300	6.5%	6.5%
}	}	}	}	}
△△市役所	36	19,250	1.3%	100.0%
累積売上比率が70%以上になるまでの契約先数				12件
店舗番号(1~9)を入力してください→9				
川崎店 【売上金額合計： 620,250】				
}	}	}	}	}

(第3図)

処理条件

1. 第1図の契約先データは契約先コードの昇順に記録されている。なお、契約先コードは、次の例のように構成されており、店舗番号は1(水戸店)~9(川崎店)であり、契約先番号は店舗ごとに1から割り振られている。

例 1 0 2 9 → 1 0 2 9
店舗番号 契約先番号

2. 第2図の売上データは日付、契約先コード、商品番号の昇順に記録されている。なお、商品番号は1(日替わり弁当)~20(かつカレー)である。

3. 配列 sName に店舗名を記憶する。なお、sName の添字は店舗番号と対応している。

配列

sName	(0)	(1)	(2)	~	(8)	(9)
		水戸店	佐野店	~	市原店	川崎店

4. 店舗情報を管理する Shop クラスをインスタンス化し、配列 sObj に記憶する。なお、sObj の添字は店舗番号と対応している。

配列

sObj	(0)	(1)	(2)	~	(8)	(9)
				~		

5. 配列 pPrice に商品単価を記憶する。なお、pPrice の添字は商品番号と対応している。

配列

pPrice	(0)	(1)	(2)	~	(19)	(20)
		500	550	~	600	650
		(日替わり弁当)	(幕の内弁当)	~	(チキンカレー)	(かつカレー)

6. 第1図の契約先データを読み、店舗番号をもとに、配列 (ArrayList) cName に契約先名を順に記憶する。なお、cName の添字は契約先番号とし、契約先名を記憶するたびに cName の要素数(データ件数)が増加する。

配列 (ArrayList)

cName	(0)	(1)	(2)	~
				~

7. 配列 pt に契約先番号を記憶する。なお、pt の添字は契約先番号と対応している。

配列

pt	(0)	(1)	(2)	~
		1	2	~

8. 第2図の売上データを読み、次の処理を行う。

・ 売上金額を次の計算式で求め、配列 sTotal に数量を、配列 aTotal に売上金額を集計する。なお、sTotal, aTotal の添字は契約先番号と対応している。また、売上金額合計を求める。

売上金額 = 数量 × 商品単価

配列

sTotal	(0)	(1)	(2)	~
				~
aTotal	(0)	(1)	(2)	~
				~

9. データを読み終えたあと、次の処理を行う。

- ・ 配列 pt を利用して配列 aTotal の降順に並べ替える。
- ・ キーボードから店舗番号が入力されたら、第3図のようにディスプレイに表示する。
- ・ 契約先ごとの売上比率を次の計算式で求める。

売上比率 = 売上金額計 × 100 ÷ 売上金額合計

・ 累積売上比率が70%以上になるまでの契約先数を求め、第3図のように表示する。なお、1件目に表示する契約先の売上比率が70%以上の場合もある。

10. キーボードから 0 が入力されたら処理を終了する。

11. Shop クラスの省略されているメソッドは次のとおりである

public void setData(String cName)	契約先名を配列 (ArrayList) cName に順に記憶する。 なお、cName は ArrayList<String> として宣言してある。
public void arrayDim()	配列 pt を定義し、初期値を記憶する。 配列 sTotal, aTotal を定義する。 なお、pt, sTotal, aTotal は、配列 (ArrayList) cName の要素数(データ件数)に合わせて定義する。

(注) ArrayList は、要素数(データ件数)の変更可能な配列である。

配列 (ArrayList).add(データ) は、データを要素として新しく追加する。

配列 (ArrayList).size() は、要素数(データ件数)を戻り値として返す。

配列 (ArrayList).get(添字) は、添字で指定された要素(データ)を戻り値として返す。

<Javaプログラム>

```

//クラスShop
import java.util.ArrayList;
public class Shop {
    private String sName;
    private ArrayList<String> cName = new ArrayList<String>();
    private int[] pt, sTotal, aTotal; private int atTotal;
    public Shop(String sName) {
        this.sName = sName;
        this.cName.add("");
    }
    } (注)省略されているメソッドの詳細は処理条件11
    public void calc(int cCode, int sales, int tSales) {
        int idx = cCode % 1000;
        (1)
        aTotal[idx] += tSales;
        atTotal += tSales;
    }
    public void sort() {
        int tmp, work;
        for(int m = 1; m < cName.size() - 1; m++) {
            tmp = m;
            for(int n = m + 1; n < cName.size(); n++) {
                if((2)) {
                    tmp = n;
                }
            }
            if(m != tmp) {
                work = pt[m]; pt[m] = pt[tmp]; pt[tmp] = work;
            }
        }
    }
    public void output() {
        System.out.printf("%-3s 【売上金額合計: %9d】 ¥n", sName, atTotal);
        double cShare = 0.0;
        int count = 0;
        aTotal[0] = 0;
        for(int r = 1; r < cName.size(); r++) {
            double share = (double) (3);
            cShare += share;
            if(cShare < 70.0) {
                count += 1;
            } else if((4) cShare >= 70.0) {
                count += 1;
                aTotal[0] = 1;
            }
            System.out.printf(" %-8s %3d %7d %4.1f% %5.1f%¥n", cName.get(pt[r]), sTotal[pt[r]], aTotal[pt[r]], share, cShare);
        }
        System.out.printf(" 累積売上比率が70%以上になるまでの契約先数 %3d件¥n", count);
    }
}
//クラスSalesAnalysis
import java.io.BufferedReader;
}
public class SalesAnalysis {
    private static String[] sName = { "", "水戸店", "佐野店", "~ "市原店", "川崎店" };
    private static Shop[] sObj = new Shop[10];
    private static int[] pPrice = { 0, 500, 550, ~ 600, 650 };
    public static void main(String[] args) {
        for(int g = 1; g < sName.length; g++) {
            sObj[g] = new Shop(sName[g]);
        }
        try {
            BufferedReader fileIn1 = new BufferedReader(new FileReader("client.csv"));
            String line;
            while((line = fileIn1.readLine()) != null) {
                String[] str = line.split(",");
                int cCode = Integer.parseInt(str[0]); String cName = str[1]; sObj[cCode / 1000].setData(cName);
            }
            fileIn1.close();
            for(int i = 1; i < sObj.length; i++) {
                sObj[i].arrayDim();
            }
            BufferedReader fileIn2 = new BufferedReader(new FileReader("selling.csv"));
            while((line = fileIn2.readLine()) != null) {
                String[] str = line.split(",");
                int cCode = Integer.parseInt(str[1]); int pCode = Integer.parseInt(str[2]); int sales = Integer.parseInt(str[3]);
                int tSales = sales * pPrice[pCode];
                (5)(cCode, sales, tSales);
            }
            fileIn2.close();
        } catch(Exception e) {
            System.out.println("エラーが発生しました" + e);
        }
        for(int i = 1; i < sObj.length; i++) {
            sObj[i].sort();
        }
        Scanner keyboardIn = new Scanner(System.in);
        System.out.print("店舗番号(1~9)を入力してください→"); int shopCode = keyboardIn.nextInt();
        while(shopCode != 0) {
            sObj[shopCode].output();
            System.out.print("店舗番号(1~9)を入力してください→"); shopCode = 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

あるファミリーレストランのメニューデータと1か月分の勤務時間データと売上データを読み、店舗別人時売上高とメニュー別集計結果を表示する。

入力データ

メニューデータ (ファイル名: menu.csv)

メニュー番号	メニュー名	単価
××	×～×	××××

(第1図)

勤務時間データ (ファイル名: kinmu.csv)

従業員コード	勤務時間
××××	×××

(第2図)

売上データ (ファイル名: uriage.csv)

日付	店舗番号	メニュー番号	区分番号	数量
××××	××	××	×	×××

(第3図)

ユーザーフォーム・実行結果

The screenshot shows a user form with the following elements:

- 店舗別人時売上高** (Store-wise person-hour sales): A table with columns for store name, working hours, and sales per person-hour.

店舗名	勤務時間計	1時間あたりの人時売上高
つくば	3,256	3,183
宇都宮	3,015	4,230
前橋	3,352	3,433
高崎	2,958	2,916
大宮	3,562	3,477
川口	3,480	3,797
千葉	2,859	3,653
船橋	3,188	3,824
八王子	3,946	3,981
横浜	3,754	4,396
- 分析項目** (Analysis items): Input fields for store number (3), store name (前橋), district number (1), and district name (イトイン).
- メニュー別集計結果** (Menu-wise summary results): A table showing menu names, total sales amount, sales ratio, and cumulative sales ratio.

メニュー名	売上金額計	売上比率	売上比率累計
マルゲリータピザ	1,533,600	14.8%	14.8%
ビーフカレー	1,025,400	9.9%	24.7%
チーズハンバーグ	842,140	8.1%	32.8%
フライドポテト	745,710	7.2%	40.0%
}	}	}	}
- 売上比率累計が70%以上になるまでのメニュー数** (Number of menus until cumulative sales ratio reaches 70% or more): 14.

(第4図)

処理条件

- 第1図のメニュー番号は1からの連番であり、メニューは50種類以下である。
- 第2図の勤務時間データは従業員コードの昇順に記録されている。なお、従業員コードは次の例のように構成されている。また、店舗番号は1(つくば)~10(横浜)であり、店舗ごとの従業員番号は1からの連番である。
例 1025 → 10 25
店舗番号 従業員番号
- 第3図の売上データは日付の昇順に記録されている。なお、区分番号は1(イトイン)と2(テイクアウト)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 Tmei に店舗名を記憶する。なお、Tmei の添字は店舗番号と対応している。

配列

Tmei (0) (1) (2) ~ (10)

	つくば	宇都宮	~	横浜
--	-----	-----	---	----

- 第1図のメニューデータを読み、配列 Mei にメニュー名を、配列 Tanka に単価を記憶する。なお、Mei, Tanka の添字はメニュー番号と対応している。

配列

Mei		Tanka	
(0)		(0)	
(1)	ハンバーグ	(1)	690
}	}	}	}
(50)		(50)	

- 第2図の勤務時間データを読み、店舗ごとに配列 Jkei に勤務時間を集計する。なお、Jkei の添字は店舗番号と対応している。

配列

Jkei (0) (1) (2) ~ (10)

			~	
--	--	--	---	--

- 第3図の売上データを読み、店舗・区分ごとに配列 Ukei に売上金額計を求める。なお、Ukei の 0 行目には店舗・区分ごとの合計を求める。また、Ukei の行方向の添字はメニュー番号と対応している。

配列

Ukei (0) (1) (2) ~ (10) (11) (12) ~ (20)

(0)			~				~		(合計)
(1)			~				~		
}	}	}	}	}	}	}	}	}	}
(50)			~				~		
	(つくば)	(宇都宮)	~	(横浜)	(つくば)	(宇都宮)	~	(横浜)	
	(イトイン)			(テイクアウト)					

- データを読み終えたあと、各店舗の1時間あたりの人時売上高を次の計算式で求め、第4図のように Label11 に店舗別人時売上高を表示する。
1時間あたりの人時売上高 = (イトインの売上金額計の合計 + テイクアウトの売上金額計の合計) ÷ 勤務時間計
- 配列 Work を利用して、店舗・区分ごとに売上金額計の降順に並べ替える。

配列

Work (0) (1) (2) ~ (10) (11) (12) ~ (20)

(0)			~				~		
(1)			~				~		
}	}	}	}	}	}	}	}	}	}
(50)			~				~		
	(つくば)	(宇都宮)	~	(横浜)	(つくば)	(宇都宮)	~	(横浜)	
	(イトイン)			(テイクアウト)					

- 第4図のように TextBox1 に店舗番号を、TextBox2 に区分番号を入力し、「分析」ボタンをクリックすると、次の処理を行う。
 - Label2 に店舗名を、Label3 に区分名を表示する。
 - 入力した店舗番号と区分番号について、売上比率を次の計算式で求め、TextBox3 にメニュー別集計結果を表示する。
売上比率 = 店舗・区分ごとの売上金額計 × 100 ÷ 店舗・区分ごとの売上金額計の合計
 - 売上比率累計が70%以上になるまでのメニュー数を求め、Label14 に表示する。なお、1件目に表示するメニューの売上比率が70%以上の場合もある。

< マクロ言語プログラム >

```

Option Explicit
Dim Tmei(10) As String, Mmei(50) As String, Ken As Long, ~ Work(50, 20) As Long

Private Sub UserForm_Initialize()
Dim Mban As Long, Mmei As String, Mtan As Long, ~ m As Long
Tmei(1) = "つくば": Tmei(2) = "宇都宮": ~ Tmei(10) = "横浜"
Call Syokika: Label1.Caption = ""
Open ThisWorkbook.Path & "¥menu.csv" For Input As #1
Do While Not EOF(1)
Input #1, Mban, Mmei, Mtan
Mei(Mban) = Mmei: Tanka(Mban) = Mtan
Loop
Close #1
Ken = Mban
Open ThisWorkbook.Path & "¥kinmu.csv" For Input As #2
Do While Not EOF(2)
Input #2, Jcode, Kji
Tban = Int(Jcode / 100)
Jkei(Tban) = Jkei(Tban) + Kji
Loop
Close #2
Open ThisWorkbook.Path & "¥uriage.csv" For Input As #3
Do While Not EOF(3)
Input #3, Hi, Tban, Mban, Kban, Su
Kin = Su * Tanka(Mban)
Soe = (Kban - 1) * 10 + Tban
(1)
Ukei(0, Soe) = Ukei(0, Soe) + Kin
Loop
Close #3
For g = 1 To 10
Nuri = (2)
Label1.Caption = Label1.Caption & Tmei(g) & " " & Format(Format(Jkei(g), "##,##0"), "#####") & " " & _
Format(Format(Nuri, "##,##0"), "#####") & Chr(13) & Chr(10)
Next g
For (3)
For h = 1 To Ken
Work(h, i) = h
Next h
For k = 1 To Ken - 1
Max = k
For m = k + 1 To Ken
If (4) Then
Max = m
End If
Next m
If k <> Max Then
Work(0, i) = Work(k, i): Work(k, i) = Work(Max, i): Work(Max, i) = Work(0, i)
End If
Next k
Next i
End Sub

Private Sub 分析_Click()
Dim p As Long, r As Long, Ritu As Double, Rui As Double, Msu As Long
TextBox3.Text = ""
Ukei(0, 0) = 0
Tban = Val(TextBox1.Text)
Label2.Caption = Tmei(Tban)
Kban = Val(TextBox2.Text)
If Kban = 1 Then
Label3.Caption = "イートイン"
Else
Label3.Caption = "テイクアウト"
End If
Soe = (Kban - 1) * 10 + Tban
For p = 1 To Ken
r = Work(p, Soe)
Ritu = Ukei(r, Soe) * 100 / Ukei(0, Soe)
Rui = Rui + Ritu
If Rui < 70 Then
Msu = Msu + 1
ElseIf (5) Rui >= 70 Then
Msu = Msu + 1
Ukei(0, 0) = 1
End If
Next p
Label4.Caption = Msu
End Sub

Private Sub クリア_Click()
Call Syokika
End Sub

Private Sub 終了_Click()
End
End Sub

Private Sub Syokika()
TextBox1.Text = "": TextBox2.Text = "": Label2.Caption = "": Label3.Caption = "": TextBox3.Text = "": Label4.Caption = ""
End Sub

```

(令和元年 9 月22日実施)

主催 公益財団法人 全国商業高等学校協会

令和元年度 (第61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	
		J a v a	マクロ言語

小 計

合 計

選択言語を で囲むこと

(令和元年 9月22日実施)

主催 公益財団法人 全国商業高等学校協会

令和元年度 (第61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	各2点 計10点
	ウ	サ	コ	カ	キ	

【2】	1	2	3	4	5	各2点 計10点
	ク	コ	エ	ケ	ア	

【3】	1	2	3	4	5	各2点 計10点
	ア	ウ	ア	イ	イ	

小 計
30

【4】	(1)	(2)	(3)	(4)	(5)	各3点 計15点
	イ	カ	オ	エ	ウ	

【5】	(1)	(2)	(3)	(4)	(5)	各3点 計15点
	エ	イ	ク	ケ	オ	

【6】	(1)	(2)	(3)	(4)	(5)	各3点 計15点
	キ	エ	ウ	ア	コ	

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) s T o t a l [i d x] += s a l e s
	(2) a T o t a l [p t [n]] > a T o t a l [p t [t m p]]
	(3) a T o t a l [p t [r]] * 1 0 0 / a t T o t a l
	(4) a T o t a l [0] == 0 &&
	(5) s O b j [c C o d e / 1 0 0 0] . c a l c

[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) U k e i (M b a n , S o e) = U k e i (M b a n , S o e) + K i n
	(2) (U k e i (0 , g) + U k e i (0 , g + 1 0)) / J k e i (g)
	(3) i = 1 T o 2 0
	(4) U k e i (W o r k (m , i) , i) > U k e i (W o r k (M a x , i) , i)
	(5) U k e i (0 , 0) = 0 A n d

各5点 計25点

小 計
25

合 計
100