

2015年 1 月18日 実施

平成26年度（第52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10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・COBOL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. コンピュータシステムの故障が復旧してから、次に故障するまでの平均時間。
2. ネットワーク上において、送信先アドレスなどの情報をもとに、データを通過させるかどうか判断するセキュリティ機能。
3. インターネット上で、電子メールを保存しているサーバからメールを受信するためのプロトコル。
4. 複数のモジュールからなるプログラムのテストを行う際に、下位モジュールから順次結合して検証するテスト。
5. 公開鍵暗号方式などを利用し、データの改ざんがないことと送信元の確認を行うしくみ。

解答群

ア. デジタル署名	イ. 単体テスト	ウ. MTBF
エ. SSD	オ. ボトムアップテスト	カ. NAS
キ. ゲートウェイ	ク. SMTP	ケ. MTTR
コ. POP	サ. トップダウンテスト	シ. パケットフィルタリング

【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. 機能テスト | 2. Cookie | 3. MIPS |
| 4. フェールソフト | 5. プロトタイプモデル | |

< B 群 >

- ア. コンピュータシステムの内部で同期をとるために、1秒間に発生する信号の回数。
- イ. Webサイトにアクセスした際、アクセスした日時や回数などの情報が、ブラウザを通じてユーザのコンピュータに一時的に保存されるしくみ。
- ウ. システムをいくつかのサブシステムに分割し、設計・プログラミング・テストなどの各工程を繰り返し行うことによって開発する手法。
- エ. コンピュータシステムやソフトウェアが、ユーザの要望した内容を満たしているかを検証するテスト。
- オ. 人間はミスをするということを前提とし、ユーザが誤った操作をしても危険な状態にならないように配慮して設計すること。
- カ. コンピュータシステムが1秒間に処理できる命令数を100万単位で表したものの。
- キ. システムに障害が発生したとき、システム全体を停止させるのではなく、障害の影響を最小限にとどめるため、正常に稼働する部分だけでシステムを運用すること。
- ク. ブラウザからの要求により、Webサーバ上でプログラムを動作させるしくみ。
- ケ. 試作品に対するユーザの評価を反映し、システムを開発する手法。
- コ. コンピュータシステムを修正した際、修正していない部分に影響があるかを検証するテスト。

【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. 10進数の3.375を2進数で表したものの。

ア. 11.0011	イ. 11.011	ウ. 11.11
------------	-----------	----------
2. システム開発において要件定義をもとに入出力画面や出力帳票を設計する工程。

ア. 外部設計	イ. 基本設計	ウ. 内部設計
---------	---------	---------
3. 同一のネットワーク内で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. ネットワークアドレス	イ. サブネットマスク	ウ. ホストアドレス
---------------	-------------	------------
4. 浮動小数点演算で絶対値が非常に近い2つの値で減算を行ったときに、有効数字の桁数が極端に少なくなること。

ア. 桁落ち	イ. 情報落ち	ウ. 丸め誤差
--------	---------	---------
5. 480Kバイト／トラック、25トラック／シリンダ、シリンダ数40,000の磁気ディスク装置の記憶容量は何Gバイトか。ただし、1Gバイト=10⁹バイトとする。

ア. 48Gバイト	イ. 480Gバイト	ウ. 768Gバイト
-----------	------------	------------

【 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 Atai に数値を昇順に記憶する。なお、データ件数は n 件であり、同じ数値はないものとする。

配列

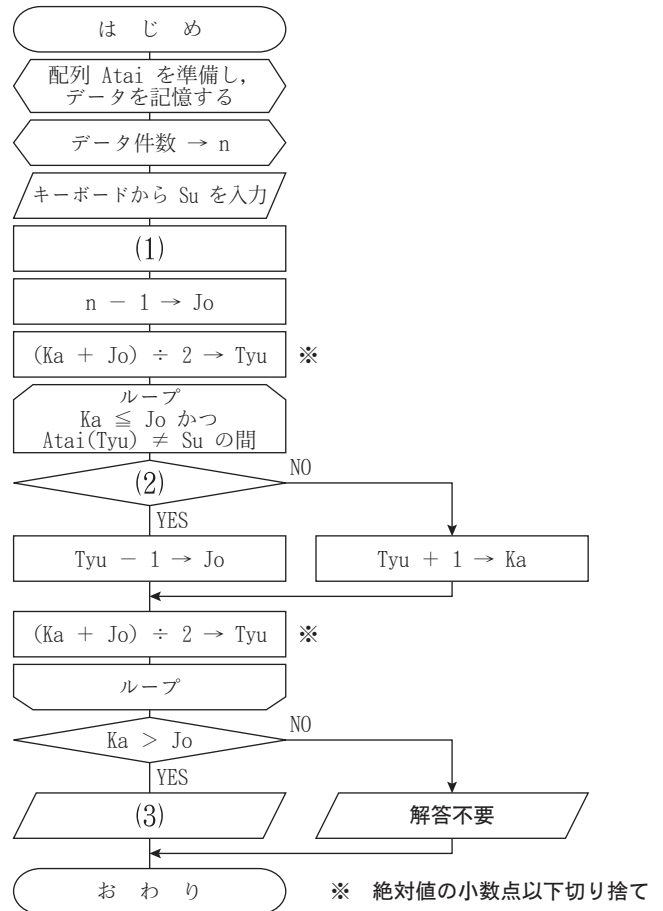
Atai	(0)	～	($n - 1$)
	101	～	228

2. キーボードから入力した数値を探索し、見つからなかった場合は「該当データなし」を、見つかった場合は「該当データあり」を表示する。

解答群

- ア. $0 \rightarrow Ka$
- イ. "該当データあり" を表示
- ウ. $1 \rightarrow Ka$
- エ. $Atai(Tyu) < Su$
- オ. "該当データなし" を表示
- カ. $Atai(Tyu) > Su$

<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 Num に数値を記憶する。なお、データ件数は n 件である。

配列

Num	(0)	～	($n - 1$)
	5	～	49

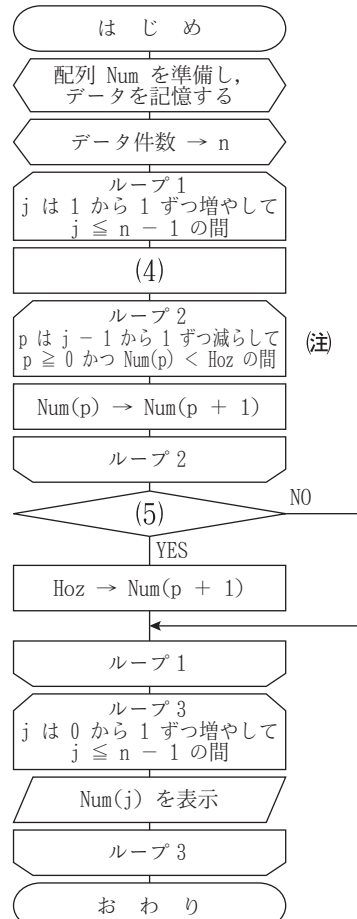
2. 配列 Num の数値を降順に並べ替える。
3. 並べ替えが終わったら、配列 Num の内容を表示する。

解答群

- ア. $Num(j - 1) \rightarrow Hoz$
- イ. $Num(j) \rightarrow Hoz$
- ウ. $p = j$
- エ. $p \neq j - 1$

(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

衣料品卸売業者の得意先データを読み、得意先別取引数集計一覧をディスプレイに表示する。

入力データ

伝票番号 (Dban)	得意先コード (Tcode)	分類コード (Bcode)	商品名 (Smei)	取引数 (Su)
××××	××	×	×～×	×××

(第 1 図)

実行結果

(得意先別取引数集計一覧)		
新宿デパート		
(分類名)	(商品名)	(取引数)
トップス	Y シャツ	1,267
	ポロシャツ	1,034
}	}	}
	(総計)	45,108
愛知百貨店		
}		

(第 2 図)

処理条件

1. 配列 Hbun に商品の分類名を、配列 Hbunsu に各分類に属する商品数を、配列 Hsmei に分類順に商品名を記憶する。なお、取扱商品は100種類である。また、第 1 図の分類コードは 1～5 であり、Hbun および Hbunsu の添字と対応している。

配列

Hbun	Hbunsu	Hsmei
(0)	(0)	(0)
(1) トップス	(1) 23	(1) Y シャツ
}	}	(2) ポロシャツ
(5) アウター	(5) 19	}
		(99) コート
		(100) ブルゾン

2. 配列 Htmei に得意先名を記憶する。なお、第 1 図の得意先コードは 1～20 であり、Htmei の添字と対応している。

配列

Htmei	(0)	(1)	～	(20)
		新宿デパート	～	大宮百貨店

3. 第 1 図の入力データを読み、商品名をもとに配列 Hsmei を探索し、配列 Hsu に取引数を集計する。なお、0 行目には総計を求める。また、Hsu の行方向の添字は Hsmei の添字と、列方向の添字は得意先コードと対応している。

配列

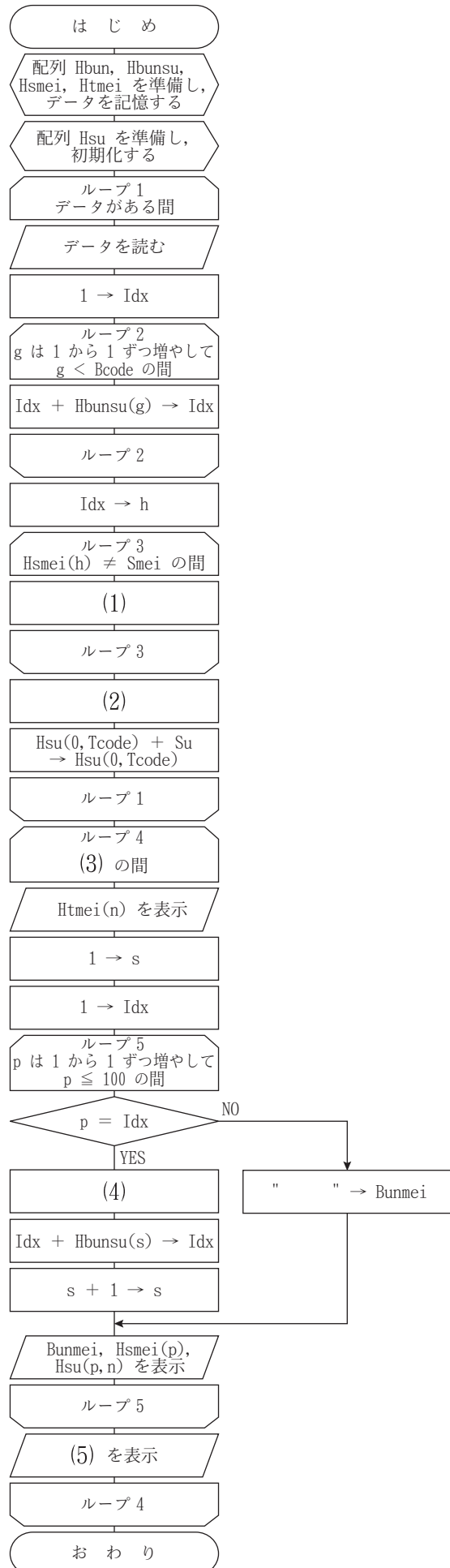
Hsu	(0)	(1)	～	(20)
(0)			～	
(1)			～	
}	}	}	}	}
(99)			～	
(100)			～	

4. 入力データが終了したら、第 2 図のように表示する。
5. データにエラーはないものとする。

解答群

- ア. n は 1 から 1 ずつ増やして $n \leq 20$
 イ. $h - 1 \rightarrow h$
 ウ. $Hsu(0, n)$
 エ. $Hbun(p) \rightarrow Bunmei$
 オ. $Hsu(h, Tcode) + Su \rightarrow Hsu(h, Tcode)$
 カ. $h + 1 \rightarrow h$
 キ. $Hsu(g, Tcode) + Su \rightarrow Hsu(g, Tcode)$
 ク. n は 0 から 1 ずつ増やして $n < 20$
 ケ. $Hsu(p, n)$
 コ. $Hbun(s) \rightarrow Bunmei$

<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

日本を訪れた外国人観光客の調査データを読み、地域別観光客数一覧をディスプレイに表示する。

入力データ

調査日 (Hi)	連番 (Ban)	地域コード (Tcd)	観光目的コード (Kcd)
××××	×××××	×	××

(第 1 図)

実行結果

(地域別観光客数一覧)		
(地域名)	(観光客数合計)	
北海道	452,906	
(主要観光目的と構成比率(％))		
グルメ	テーマパーク	伝統芸能
31.4	25.2	13.1
東北		
}		

(第 2 図)

処理条件

1. 配列 Tiiki に地域名を、配列 Kanko に観光目的を記憶する。なお、第 1 図の地域コードは 1～6 であり、Tiiki の添字と、観光目的コードは 1～10 であり、Kanko の添字と対応している。

配列

Tiiki	(0)	(1)	～	(6)
		北海道	～	九州・沖縄
Kanko	(0)	(1)	～	(10)
		グルメ	～	買物

2. 第 1 図の入力データを読み、観光客数を配列 Kei に集計する。なお、11 列目には観光客数合計を求める。また、Kei の行方向の添字は地域コードと、列方向の添字は観光目的コードと対応している。

配列

Kei	(0)	(1)	～	(10)	(11)
(0)			～		
(1)			～		
}	}	}	}	}	}
(6)			～		

(観光客数合計)

3. 入力データが終了したら、次の処理を行う。
- 第 2 図のように地域名と観光客数合計を表示する。
 - 地域名ごとに、観光目的を観光客数の降順に並べ替え、観光客数の構成比率を求める。なお、観光目的の並べ替え用に配列 Hoz を利用する。

配列

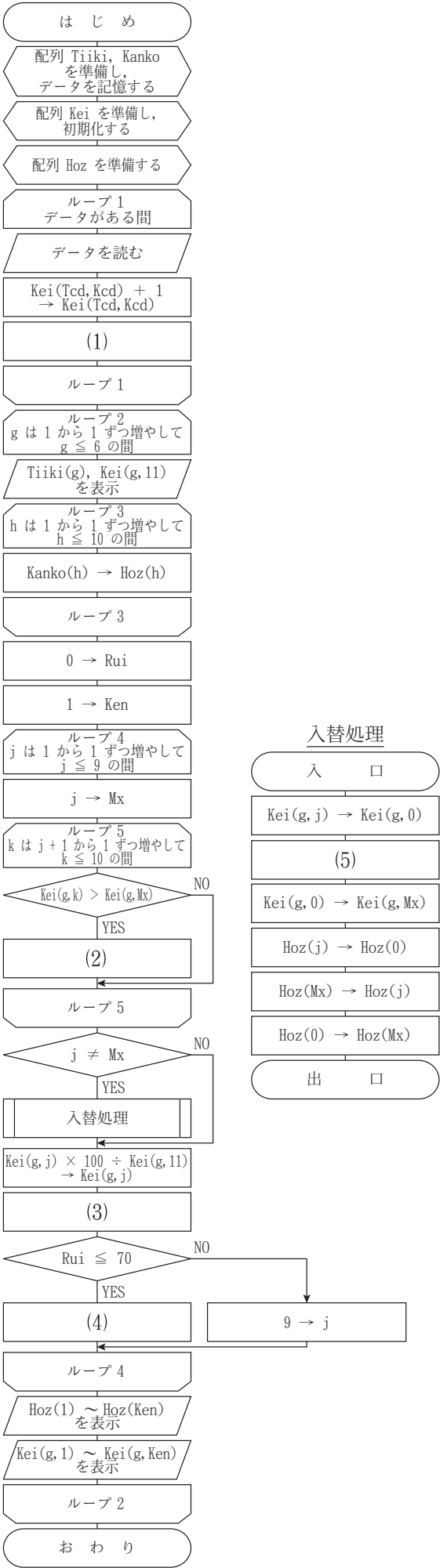
Hoz	(0)	(1)	～	(10)
			～	

- 第 2 図のように観光客数の降順に、観光目的とその構成比率を表示する。ただし、表示する観光目的は構成比率の累計が 70% 以内のものとする。
4. データにエラーはないものとする。

解答群

- ア. $Kei(0, Kcd) + 1 \rightarrow Kei(0, Kcd)$
イ. $Rui + Kei(g, j) \rightarrow Rui$
ウ. $g \rightarrow Ken$
エ. $j \rightarrow Ken$
オ. $Rui + 1 \rightarrow Rui$
カ. $g \rightarrow Mx$
キ. $Kei(Tcd, 11) + 1 \rightarrow Kei(Tcd, 11)$
ク. $Kei(g, Mx) \rightarrow Kei(g, j)$
ケ. $k \rightarrow Mx$
コ. $Kei(g, j) \rightarrow Kei(g, Mx)$

<流れ図>



【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

売上データを読み、指定した店舗コードの集計結果を表示する。

入力データ

売上データ（ファイル名：selling.csv）

店舗コード	売上番号	スイーツコード	数量	販売種別
××	×××	××	×××	×

(第1図)

実行結果

店舗コード(1～10)を入力してください。→5

横浜店

(スイーツ名)	(売上金額計)	(順位)	(平均との差)
夜空の星大福	33,000円	14位	-4,620円
まるごとみかん大福	41,760円	8位	3,944円
栗あんみつ	93,000円	1位	23,100円
}	}	}	}
宇治抹茶ロールケーキ	36,800円	11位	5,290円
黒胡麻カステラ	14,400円	26位	-6,240円
葛餅詰め合わせ	28,000円	16位	-29,960円
売上金額合計	1,028,240円		
地方発送件数	9件		

店舗コード(1～10)を入力してください。→10

福岡店

}

(第2図)

処理条件

1. 店舗は10店舗あり、配列 name に店舗名を記憶する。なお、第1図の店舗コードは 1～10 であり、name の添字と対応している。

配列

name	(0)	(1)	(2)	～	(9)	(10)
		札幌店	仙台店	～	広島店	福岡店

2. 店舗情報を管理する ShopInfo クラスをインスタンス化し、配列 shop に記憶する。なお、配列 name, shop は添字で対応している。

配列

shop	(0)	(1)	(2)	～	(9)	(10)
				～		

3. スイーツは30種類あり、配列 sweetsName にスイーツ名、配列 sweetsPrice に単価を記憶する。なお、第1図のスイーツコードは 1～30 であり、sweetsName, sweetsPrice, 配列 sweetsTotal, 配列 total, 配列 rank の添字と対応している。また、sweetsName, sweetsPrice, sweetsTotal は static で定義されている。

配列

sweetsName	(0)	(1)	～	(29)	(30)
		夜空の星大福	～	黒胡麻カステラ	葛餅詰め合わせ
sweetsPrice	(0)	(1)	～	(29)	(30)
		2200	～	2400	2800
sweetsTotal	(0)	(1)	～	(29)	(30)
			～		
total	(0)	(1)	～	(29)	(30)
			～		
rank	(0)	(1)	～	(29)	(30)
			～		

4. 第1図の売上データを読み、次の処理を行う。
- スイーツコードをもとに、数量と単価から売上金額を求め、店舗別に配列 total に集計する。なお、配列 sweetsTotal に合計を求める。
 - 販売種別は 1（通常販売）または、2（地方発送）であり、販売種別が 2 の場合、地方発送件数を集計する。
5. データを読み終えたあと、配列 rank を利用して店舗別に売上金額計の降順に順位をつける。
6. 第2図のようにキーボードから店舗コードを入力すると、該当の店舗名と各スイーツ名、売上金額計、順位、平均との差をディスプレイに表示する。なお、売上金額合計、地方発送件数を表示する。
7. 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ShopInfo
public class ShopInfo {
    private static String[] sweetsName = { "", "夜空の星大福", "黒胡麻カステラ", "葛餅詰め合わせ" };
    private static int[] sweetsPrice = { 0, 2200, ~ 2400, 2800 };
    private static int[] sweetsTotal = { 0, 0, ~ 0, 0 };
    private String name;
    private int shopTotal = 0;
    private int localShipment = 0;
    private int[] total = new int[31];
    private int[] rank = new int[31];
    public ShopInfo(String name) {
        this.name = name;
        for(int j = 1; j < 31; j++) {
            total[j] = 0;
            (1);
        }
    }
    public void salesTotal(int sweetsCode, int quantity, int kind) {
        int sales = (2);
        total[sweetsCode] += sales;
        sweetsTotal[sweetsCode] += sales;
        shopTotal += sales;
        if(kind == 2) {
            localShipment += 1;
        }
    }
    public void ranking() {
        for(int m = 1; m < 30; m++) {
            for(int n = m + 1; n < 31; n++) {
                if(( 解答不要 ) {
                    rank[m] += 1;
                } else if( (3) ) {
                    rank[n] += 1;
                }
            }
        }
    }
    public void outName() {
        System.out.println(name);
    }
    public void outDetails(int j) {
        int difference = total[j] - sweetsTotal[j] / 10;
        System.out.printf("¥t%-10s ¥t%, 9d円 ¥t%2d位 ¥t%, 7d円¥n", sweetsName[j], total[j], rank[j], difference);
    }
    public void outTotal() {
        System.out.printf("¥t 売上金額合計 ¥t%, 9d円¥n", shopTotal);
        System.out.printf("¥t 地方発送件数 ¥t%, 9d件¥n", localShipment);
    }
}

//クラスSalesAnalysis
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Scanner;

public class SalesAnalysis {
    private static String[] name = { "", "札幌店", "仙台店", "銀座店", "新宿店", "横浜店", "名古屋店", "大阪店", "京都店", "広島店", "福岡店" };
    private static ShopInfo[] shop = new ShopInfo[11];
    public static void main(String[] args) {
        try {
            for(int i = 1; i < 11; i++) {
                shop[i] = new ShopInfo((4));
            }
            BufferedReader fileIn = new BufferedReader(new FileReader("selling.csv"));
            String line;
            while((line = fileIn.readLine()) != null) {
                String[] str = line.split(",");
                int code = Integer.parseInt(str[0]);
                int sweetsCode = Integer.parseInt(str[2]);
                int quantity = Integer.parseInt(str[3]);
                int kind = Integer.parseInt(str[4]);
                shop[code].salesTotal(sweetsCode, quantity, kind);
            }
            fileIn.close();
        } catch(Exception e) {
            System.out.println("エラーが発生しました" + e);
        }
        for(int i = 1; i < 11; i++) {
            shop[i].ranking();
        }
        Scanner keyboardIn = new Scanner(System.in);
        System.out.print("店舗コード(1~10)を入力してください。→");
        int code = keyboardIn.nextInt();
        while(code != 0) {
            shop[code].outName();
            for(int j = 1; j < 31; j++) {
                (5);
            }
            shop[code].outTotal();
            System.out.print("店舗コード(1~10)を入力してください。→");
            code = keyboardIn.nextInt();
        }
        keyboardIn.close();
    }
}
```


＜プログラムの説明＞

キャラクター関連商品の売上データを読み、商品売上一覧とキャラクター別一覧を表示する。

ユーザーフォーム・実行結果

売上データ (ファイル名: uriage.csv)

(第 1 図)

(第2図)

1. 第1図の商品コードは、次の例のようにキャラクター番号と品種番号で構成されている。なお、キャラクター番号は1～5の5種類、品種番号は1～10の10種類であり、商品は50種類である。

キャラクター番号 品種番号

2. ユーザーフォーム初期化時に、次の処理を行う。

- 配列 Kmei にキャラクター名, 配列 Hmei に品種名, 配列 Tanka に単価を記憶する。なお, Kmei の添字はキャラクター番号と対応し, Hmei, Tanka の添字は品種番号と対応している。

	Hmei		Tanka
(0)		(0)	
(1)	ボールペン	(1)	150
}	}	}	}
(10)	タイマー	(10)	800
	(品種名)		(単価)

- 第 1 図の売上データを読み、売上数量と単価から売上金額を求め、配列 `Suri` に集計する。なお、`Suri` の行方向の添字は品種番号と、列方向の添字はキャラクター番号と対応している。

Suri	(0)	(1)	$\sim$	(5)
(0)			$\sim$	
(1)			$\sim$	
$\wr$	$\wr$	$\wr$	$\wr$	$\wr$
(10)			$\sim$	

- 配列 Sjuni を利用し、配列 Suri(1, 1)～Suri(10, 5) に集計した50種類全ての商品を対象に、売上金額計の降順に順位をつける。なお、Siuni の添字は Suri の添字と対応している。

Sjuni	(0)	(1)	$\sim$	(5)
(0)			$\sim$	
(1)			$\sim$	
$\wr$	$\wr$	$\wr$	$\wr$	$\wr$
(10)			$\sim$	

- 第2図のように TextBox1 に品種名, キャラクター名, 売上金額計, 順位を表示する。
3. 第2図のように TextBox2 にキャラクター番号を入力し, 「処理」ボタンをクリックすると, 次の処理を行う。
 - TextBox3 にキャラクター名を表示する。
 - TextBox4 に該当するキャラクターの品種名, 売上金額計, 順位を表示する。

< マクロ言語プログラム >

```
Option Explicit
Dim Kmei(5) As String, Hmei(10) As String, Suri(10, 5) As Long, Sjuni(10, 5) As Long

Private Sub UserForm_Initialize()
    Dim Tanka(10) As Long, j As Long, k As Long, Hiduke As Long, Code As Long, Su As Long
    Dim Hin As Long, Kya As Long, m As Long, n As Long, G1 As Long, R1 As Long, G2 As Long, R2 As Long
    TextBox1.Text = "": TextBox2.Text = "": TextBox3.Text = "": TextBox4.Text = ""
    Kmei(1) = "しまもん": Kmei(2) = "ぜんぞうくん": Kmei(3) = "ニャボン": Kmei(4) = "いももち": Kmei(5) = "ワカビー"
    Hmei(1) = "ボールペン": ~ Hmei(10) = "タイマー"
    Tanka(1) = 150: ~ Tanka(10) = 800
    Open ThisWorkbook.Path & "Yuriage.csv" For Input As #1
    For j = 1 To 10
        For k = 1 To 5
            Suri(j, k) = 0
            (1)
        Next k
    Next j
    Do While Not EOF(1)
        Input #1, Hiduke, Code, Su
        Hin = Code Mod 100
        Kya = Int(Code / 100)
        Suri(Hin, Kya) = (2)
    Loop
    Close #1
    For m = 1 To 49
        G1 = Int((m - 1) / 5) + 1
        R1 = m - (G1 - 1) * 5
        For n = m + 1 To 50
            G2 = Int((n - 1) / 5) + 1
            R2 = n - (G2 - 1) * 5
            If (3) Then
                Sjuni(G1, R1) = Sjuni(G1, R1) + 1
            ElseIf (3) Then
                Sjuni(G2, R2) = Sjuni(G2, R2) + 1
            End If
        Next n
    Next m
    For j = 1 To 10
        For k = 1 To 5
            TextBox1.Text = TextBox1.Text & (4) & " " & Kmei(k) & " " & Format(Suri(j, k), "###,##0") & " " & _
                Format(Sjuni(j, k), "#0") & Chr(13) & Chr(10)
        Next k
    Next j
End Sub

Private Sub 処理_Click()
    Dim Kban As Long, p As Long
    TextBox4.Text = ""
    Kban = Val(TextBox2.Text)
    TextBox3.Text = Kmei(Kban)
    For (5)
        TextBox4.Text = TextBox4.Text & Hmei(p) & " " & Format(Suri(p, Kban), "###,##0") & " " & _
            Format(Sjuni(p, Kban), "#0") & Chr(13) & Chr(10)
    Next p
End Sub

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

【 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

ある旅館の 1 か月の宿泊ファイルを読み、宿泊状況一覧を印字する。

入力データ

宿泊ファイル（ファイル名：SYUKUHAKU-F、レコード名：SYUKUHAKU-R）

日 (HI) ××	宿泊人数				
	特別室 A (NIN(1))	特別室 B (NIN(2))	～	和洋室 B (NIN(8))	和洋室 C (NIN(9))
××	×	×		×	×

(第 1 図)

実行結果

（ファイル名：ITIRAN-F、レコード名：ITIRAN-R）

(宿泊状況一覧)								
(日)	(特別室 A)	(特別室 B)	～	(和洋室 B)	(和洋室 C)	(日別合計)	(定員稼働率(%))	(順位)
1	3	2	～	0	4	21	55.3	28
}	}	}	}	}	}	}	}	}
31	2	2	～	6	5	23	60.5	12
(客室別合計)	76	58	～	130	111	767	65.1	
(客室名)(客室別合計)(定員計)(定員稼働率(%))								
特別室 A	76	124		61.3				
}	}	}		}				
和洋室 C	111	186		59.7				

(第 2 図)

処理条件

- 第 1 図の宿泊ファイルのデータ件数は月の日数分あり、日の昇順に記録されている。なお、客室は 9 室である。
- テーブル HEYA-TBL には客室名が、テーブル TEI-TBL には定員があらかじめ記憶されている。なお、HEYA-TBL と TEI-TBL は添字で対応している。

テーブル HEYA-TBL

HEYA	(1)	(2)	～	(8)	(9)
	特別室 A	特別室 B	～	和洋室 B	和洋室 C

テーブル TEI-TBL

TEI	(1)	(2)	～	(8)	(9)	(10)
	4	2	～	6	6	38
(定員合計)						

- 宿泊ファイルを読み、テーブル SYU-TBL に、宿泊人数を集計する。なお、10列目には日別合計を、32行目には客室別合計を求める。また、SYU-TBL の行方向は日と、列方向はテーブル HEYA-TBL と添字で対応している。

テーブル SYU-TBL

SYU	(1)	(2)	～	(8)	(9)	(10)
(1)			～			
}	}	}	}	}	}	}
(31)			～			
(32)			～			
(客室別合計)						
(日別合計)						

- ファイルを読み終えたあと、テーブル KAD-TBL に日ごとの定員稼働率(%)を求め、テーブル JUNI-TBL に定員稼働率(%)の降順に順位をつけ、第 2 図のように印字する。

テーブル KAD-TBL

KAD	
(1)	
}	}
(31)	
(32)	

テーブル JUNI-TBL

JUNI	
(1)	
}	}
(31)	

- 最後に客室ごとの 1 か月の定員稼働率(%)を求め、第 2 図のように印字する。

＜COBOLプログラム＞

WORKING-STORAGE SECTION.

```
01 END-FLG          PIC X(01) VALUE LOW-VALUE.
   }
01 MEISAI1.
   02                PIC X(08).
   02 M-HI           PIC Z9.
   02 MEI-SYU OCCURS 10.
       03            PIC X(05).
       03 M-SYU      PIC ZZ9.
   02                PIC X(10).
   02 M-KAD1         PIC ZZ9.9.
   02                PIC X(08).
   02 M-JUNI         PIC Z9.
```

```

01  MEISAI2.
02          PIC X(02) VALUE SPACE.
02  M-HEYA   PIC X(08).
02          PIC X(05) VALUE SPACE.
02  M-SYUKEI PIC ZZ9.
02          PIC X(05) VALUE SPACE.
02  M-TEI    PIC ZZ9.
02          PIC X(07) VALUE SPACE.
02  M-KAD2   PIC ZZ9.9.

PROCEDURE DIVISION.
P1. OPEN INPUT SYUKUHAKU-F OUTPUT ITIRAN-F
    INITIALIZE SYU-TBL MEISAI1
    PERFORM UNTIL END-FLG = HIGH-VALUE
        READ SYUKUHAKU-F
        AT END
            MOVE HIGH-VALUE TO END-FLG
        NOT AT END
            PERFORM VARYING G FROM 1 BY 1 UNTIL G > 9
                MOVE NIN(G) TO SYU(HI G)
                COMPUTE SYU(HI 10) = SYU(HI 10) + NIN(G)
                COMPUTE SYU(32 G) = (1)
                COMPUTE SYU(32 10) = SYU(32 10) + NIN(G)
            END-PERFORM
            MOVE HI TO HOZON
        END-READ
    END-PERFORM
    PERFORM VARYING H FROM 1 BY 1 UNTIL H > HOZON
        COMPUTE KAD(H) ROUNDED = SYU(H 10) * 100 / TEI(10)
        (2)
    END-PERFORM
    COMPUTE KAD(32) ROUNDED = SYU(32 10) * 100 / (TEI(10) * HOZON)
    PERFORM VARYING I FROM 1 BY 1 UNTIL I > HOZON - 1
        COMPUTE J = I + 1
        PERFORM VARYING K FROM J BY 1 UNTIL K > HOZON
            IF (3) 解答不要
                THEN
                    COMPUTE JUNI(I) = JUNI(I) + 1
                ELSE
                    IF (3)
                        THEN
                            COMPUTE JUNI(K) = JUNI(K) + 1
                        END-IF
                    END-IF
            END-IF
        END-PERFORM
    END-PERFORM
    PERFORM VARYING M FROM 1 BY 1 UNTIL M > HOZON
        (4)
        PERFORM VARYING N FROM 1 BY 1 UNTIL N > 10
            MOVE SYU(M N) TO M-SYU(N)
        END-PERFORM
        MOVE KAD(M) TO M-KAD1
        MOVE JUNI(M) TO M-JUNI
        WRITE ITIRAN-R FROM MEISAI1 AFTER 1
    END-PERFORM
    MOVE SPACE TO MEISAI1
    PERFORM VARYING P FROM 1 BY 1 UNTIL P > 10
        MOVE SYU(32 P) TO M-SYU(P)
    END-PERFORM
    MOVE KAD(32) TO M-KAD1
    WRITE ITIRAN-R FROM MEISAI1 AFTER 1
    INITIALIZE KAD-TBL
    PERFORM (5)
        COMPUTE TEIKEI = TEI(S) * HOZON
        COMPUTE KAD(S) ROUNDED = SYU(32 S) * 100 / TEIKEI
        MOVE HEYA(S) TO M-HEYA
        MOVE SYU(32 S) TO M-SYUKEI
        MOVE TEIKEI TO M-TEI
        MOVE KAD(S) TO M-KAD2
        WRITE ITIRAN-R FROM MEISAI2 AFTER 1
    END-PERFORM
    CLOSE SYUKUHAKU-F ITIRAN-F
    STOP RUN.

```

(平成27年 1 月18日実施)

主催 公益財団法人 全国商業高等学校協会

平成26年度（第52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] ・ [C O B O L]

【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	COBOL

小 計

合 計

選択言語を で囲むこと

(平成27年 1 月18日実施)

主催 公益財団法人 全国商業高等学校協会

平成26年度（第52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	小計 30
	ウ	シ	コ	オ	ア		
【2】	1	2	3	4	5	各2点 計10点	
	エ	イ	カ	キ	ケ		
【3】	1	2	3	4	5	各2点 計10点	
	イ	ア	ウ	ア	イ		
【4】	(1)	(2)	(3)	(4)	(5)	各3点 計15点	
	ア	カ	オ	イ	エ		
【5】	(1)	(2)	(3)	(4)	(5)	各3点 計15点	
	カ	オ	ア	コ	ウ		
【6】	(1)	(2)	(3)	(4)	(5)	各3点 計15点	小計 45
	キ	ケ	イ	エ	ク		

..... [J a v a] ・ [マクロ言語] ・ [COBOL]

[J a v a] (注) = , 演算子の前後の空白は問わない。

【7】	<table><tr><td>(1)</td><td>r a n k [j] = 1</td></tr><tr><td>(2)</td><td>q u a n t i t y * s w e e t s P r i c e [s w e e t s C o d e]</td></tr><tr><td>(3)</td><td>t o t a l [m] > t o t a l [n]</td></tr><tr><td>(4)</td><td>n a m e [i]</td></tr><tr><td>(5)</td><td>s h o p [c o d e] . o u t D e t a i l s (j)</td></tr></table>	(1)	r a n k [j] = 1	(2)	q u a n t i t y * s w e e t s P r i c e [s w e e t s C o d e]	(3)	t o t a l [m] > t o t a l [n]	(4)	n a m e [i]	(5)	s h o p [c o d e] . o u t D e t a i l s (j)
(1)	r a n k [j] = 1										
(2)	q u a n t i t y * s w e e t s P r i c e [s w e e t s C o d e]										
(3)	t o t a l [m] > t o t a l [n]										
(4)	n a m e [i]										
(5)	s h o p [c o d e] . o u t D e t a i l s (j)										

[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】	<table><tr><td>(1)</td><td>S j u n i (j , k) = 1</td></tr><tr><td>(2)</td><td>S u r i (H i n , K y a) + S u * T a n k a (H i n)</td></tr><tr><td>(3)</td><td>S u r i (G 1 , R 1) > S u r i (G 2 , R 2)</td></tr><tr><td>(4)</td><td>H m e i (j)</td></tr><tr><td>(5)</td><td>p = 1 T o 1 0</td></tr></table>	(1)	S j u n i (j , k) = 1	(2)	S u r i (H i n , K y a) + S u * T a n k a (H i n)	(3)	S u r i (G 1 , R 1) > S u r i (G 2 , R 2)	(4)	H m e i (j)	(5)	p = 1 T o 1 0
(1)	S j u n i (j , k) = 1										
(2)	S u r i (H i n , K y a) + S u * T a n k a (H i n)										
(3)	S u r i (G 1 , R 1) > S u r i (G 2 , R 2)										
(4)	H m e i (j)										
(5)	p = 1 T o 1 0										

[COBOL]

【7】	<table><tr><td>(1)</td><td>S Y U (3 2 G) + N I N (G)</td></tr><tr><td>(2)</td><td>M O V E 1 T O J U N I (H)</td></tr><tr><td>(3)</td><td>K A D (I) > K A D (K)</td></tr><tr><td>(4)</td><td>M O V E M T O M - H I</td></tr><tr><td>(5)</td><td>V A R Y I N G S F R O M 1 B Y 1 U N T I L S > 9</td></tr></table>	(1)	S Y U (3 2 G) + N I N (G)	(2)	M O V E 1 T O J U N I (H)	(3)	K A D (I) > K A D (K)	(4)	M O V E M T O M - H I	(5)	V A R Y I N G S F R O M 1 B Y 1 U N T I L S > 9
(1)	S Y U (3 2 G) + N I N (G)										
(2)	M O V E 1 T O J U N I (H)										
(3)	K A D (I) > K A D (K)										
(4)	M O V E M T O M - H I										
(5)	V A R Y I N G S F R O M 1 B Y 1 U N T I L S > 9										

各5点 計25点

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	COBOL	25	100

選択言語を ☐ で囲むこと