プログラミング部門１級

2014年 1 月19日 実施

平成25年度（第50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Ｊａｖａ・マクロ言語・COBOLのいずれか１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級　共通問題］

【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．大量のデータやアクセスをコンピュータシステムに処理させることによって，その反応を調べるテスト。
２．主記憶装置とハードディスク装置の間に設置し，データ転送を効率よくさせるための記憶装置。
３．コンピュータシステムの総合的な評価に用いられるチェック項目の頭文字をとったもの。
４．通信ネットワークやコンピュータシステムなどが，一定時間内に処理する情報量や仕事量。
５．音声信号をディジタル信号に変換し，ＴＣＰ／ＩＰを利用して音声通話を行う技術。

解答群

ア．ＲＡＳＩＳ	**イ**．フェールセーフ	**ウ**．退行テスト
エ．レスポンスタイム	**オ**．負荷テスト	**カ**．ＶｏＩＰ
キ．キャッシュメモリ	**ク**．ＨＴＴＰＳ	**ケ**．スループット
コ．シンクライアント	**サ**．ディスクキャッシュ	**シ**．ターンアラウンドタイム

【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．スパイラルモデル　　２．ＦＴＰ　　３．ＮＡＳ
　　４．ハブ　　５．アクセスログ

＜Ｂ群＞

ア．クライアントからメールサーバへ，またはメールサーバ間で電子メールを送信するためのプロトコル。
イ．ネットワーク利用状況の把握や，不正侵入の分析をするためなどに用いられ，コンピュータ間の接続履歴を記録したもの。
ウ．コンピュータにハードウェアを接続すると，デバイスドライバの組み込みや設定を自動で行う機能。
エ．Ｗｅｂページの更新やファイルのダウンロードなどに用いられ，相手を認証することによってファイルの転送を行うプロトコル。
オ．アプリケーション障害時などに原因を調査するために用いられ，コンピュータシステムの動作状況を記録したもの。
カ．基本設計からテストまでの各工程を，後戻りしないことを前提として順番に開発を行う手法。
キ．ＬＡＮケーブルにより接続された複数のネットワーク機器間で，送受信される電気信号の衰えを補正して中継する装置。
ク．ファイルサーバとしての機能を持ち，ネットワークに接続できる大容量の補助記憶装置。
ケ．プロトコルの異なるネットワーク間において，プロトコルを変換することで接続させる装置。
コ．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を進める手法。

【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４Ｄを８ビットの２の補数で表したもの。

ア．01001101　　**イ**．10110010　　**ウ**．10110011

２．セキュリティ確保のため，組織内部のネットワークと外部のネットワーク間に設けられたネットワーク領域。

ア．ＤＮＳ　　**イ**．ＤＭＺ　　**ウ**．ＸＭＬ

３．10進数の各桁を２進数の４桁ずつにして表現したもの。

ア．浮動小数点形式　　**イ**．固定小数点形式　　**ウ**．２進化10進数

４．２つの入力が「０」と「０」のときだけ「０」を出力する論理回路。

ア．ＸＯＲ回路　　**イ**．ＯＲ回路　　**ウ**．ＡＮＤ回路

５．10,000回転／分のハードディスク装置の平均回転待ち時間を求めなさい。

ア．３ミリ秒　　**イ**．６ミリ秒　　**ウ**．10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

入力データを読み，コードごとに数値を集計し表示する。

入力データ

コード (Cod) ×××	数値 (Su) ×～×

実行結果

（コード）	（数値合計）
101	2,534
≀	≀
121	3,211
（総合計）	98,619

処理条件

1．入力データは，コードの昇順に記録されている。

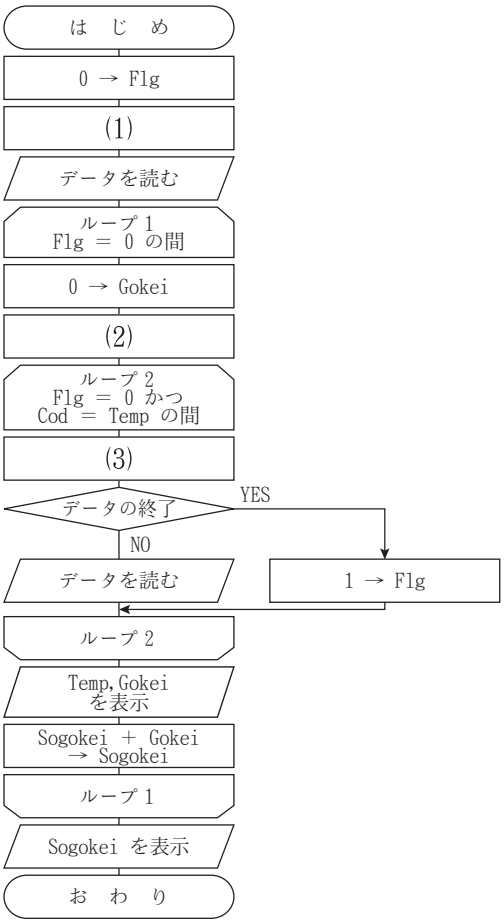
2．入力データを読み，コードごとに数値合計を集計する。なお，コードがかわるごとに，コードと数値合計をディスプレイに表示するとともに，総合計を集計する。

3．最後に総合計をディスプレイに表示する。

解答群

ア．Su → Temp
イ．Sogokei ＋ Su → Sogokei
ウ．Cod → Temp
エ．0 → Su
オ．Gokei ＋ Su → Gokei
カ．0 → Sogokei

＜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表示する。

処理条件

1．配列 Dat に数値を記憶する。なお，データ件数は n 件である。

配列

Dat	(0)	～	(n － 1)
	76	～	20

2．配列 Dat の数値を昇順に並べ替える。

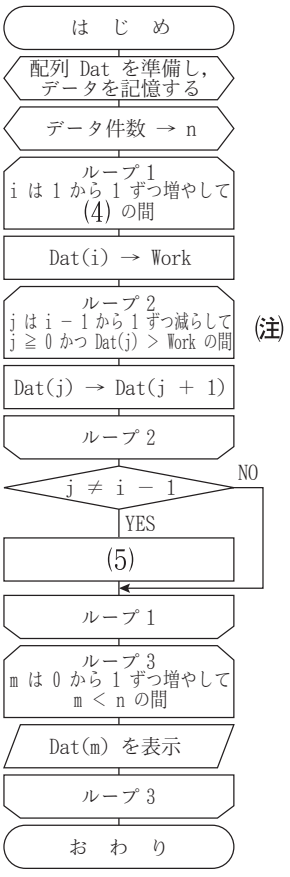
3．並べ替えが終わったら，配列 Dat の内容をディスプレイに表示する。

解答群

ア．Work → Dat(j ＋ 1)
イ．i ≦ n
ウ．Work → Dat(j)
エ．i ≦ n － 1

㈱　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

百貨店の１日分の販売金額データを読み，販売金額区分ごとに件数を集計し，時間帯別度数分布と販売金額計降順リストを表示する。

入力データ

レジ番号 (Rban)	販売時刻 (Jikoku)	販売金額 (Hkin)
××××	××××	×××××××

（第１図）

実行結果

（時間帯別度数分布）

（販売金額区分）	（10時台）	（11時台）	～	（19時台）	（20時台）
1円～ 499円	×，×××	×，×××	～	×，×××	×，×××
500円～ 999円	×，×××	×，×××	～	×，×××	×，×××
1,000円～ 1,999円	×，×××	×，×××	～	×，×××	×，×××
≀	≀	≀	≀	≀	≀
10,000円～19,999円	×，×××	×，×××	～	×，×××	×，×××
20,000円～49,999円	×，×××	×，×××	～	×，×××	×，×××
50,000円～	×，×××	×，×××	～	×，×××	×，×××

（販売金額計降順リスト）

（時間帯）	（販売金額計）
12時台	×，×××，×××
15時台	×，×××，×××
≀	≀
20時台	×，×××，×××
10時台	×，×××，×××

（第２図）

処理条件

１．百貨店の営業時間は，午前10時00分から午後８時59分であり，第１図の入力データの販売時刻は，時分の形式で次のように記録されている。

例　午後３時２３分　→　１５２３

２．第１図の入力データを読み，販売金額区分および時間帯別に件数を配列 Dosu に集計する。なお，０行目には時間帯別の販売金額計を集計する。また，０列目には販売金額区分の上限を，１～11列目には０をあらかじめ記憶する。

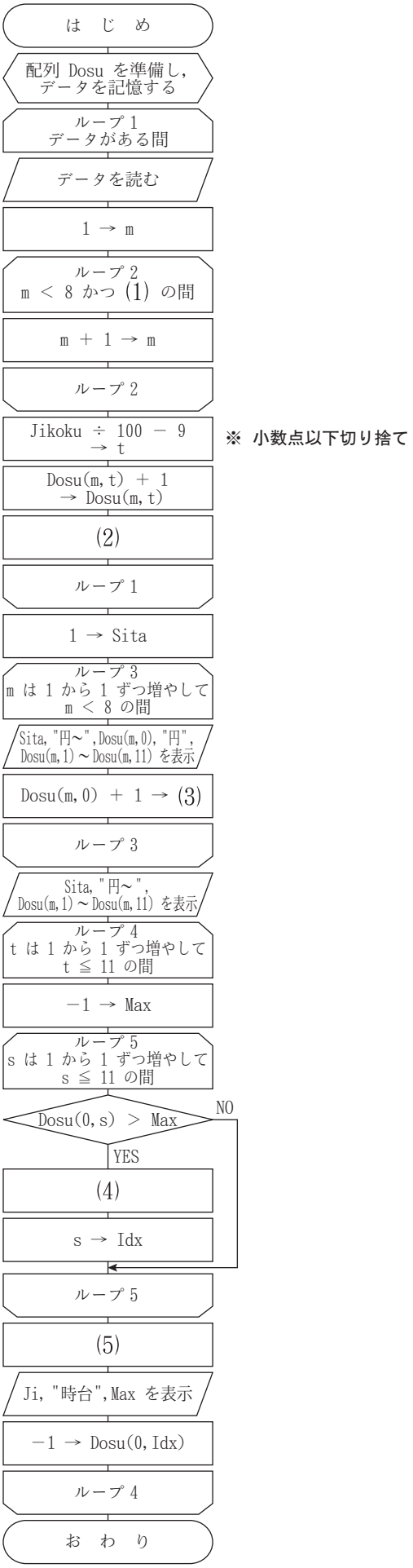
配列

Dosu	(0)	(1)	(2)	～	(10)	(11)	
(0)		0	0	～	0	0	（販売金額計）
(1)	499	0	0	～	0	0	
(2)	999	0	0	～	0	0	
(3)	1999	0	0	～	0	0	
≀	≀	≀	≀	≀	≀	≀	（件数）
(6)	19999	0	0	～	0	0	
(7)	49999	0	0	～	0	0	
(8)	0	0	0	～	0	0	
	（上限）	（10時台）	（11時台）	～	（19時台）	（20時台）	

３．入力データが終了したら，第２図のように時間帯別度数分布と販売金額計降順リストをディスプレイに表示する。なお，販売金額計が同じ場合，時間帯の昇順とする。

４．データにエラーはないものとする。

＜流れ図＞



解答群

ア．Dosu(0,t) ＋ Hkin → Dosu(0,t)
イ．Idx
ウ．Idx ＋ 9 → Ji
エ．Dosu(m,0) ＜ Hkin
オ．Dosu(0,t) ＋ 1 → Dosu(0,t)
カ．s → Max
キ．Jikoku → Ji
ク．Dosu(0,s) → Max
ケ．Dosu(m,0) ≧ Hkin
コ．Sita

【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

１か月分の入力データを読み，栄養指導資料一覧を表示する。

入力データ

日付 (Hi) ××	社員番号 (Sban) ×××	食事区分 (Sku) ××	料理コード (Rcode) ×××

（第１図）

実行結果

（栄養指導資料一覧（１日平均））

(社員名)	(性別)	(たんぱく質(g))	(脂質(g))	～	(ｺﾚｽﾃﾛｰﾙ(mg)) (男性：2,300kcal以上) (女性：1,800kcal以上)	(ｶﾛﾘｰ(kcal))
佐藤　▽▽	男	164.2	231.9	～	923.1	4,082.7
≀	≀	≀	≀	≀	≀	≀
高橋　□□	女	79.4	109.2	～	344.0	1,882.0

（第２図）

処理条件

１．配列 Syain に社員名と性別を，配列 Eiyo に各料理１食分の栄養摂取量とカロリーを記憶する。なお，Syain の行方向の添字は社員番号と，Eiyo の行方向の添字は料理コードと対応している。

配列

Syain	(0)	(1)
(0)		
(1)	新井　○○	男
≀	≀	≀
(100)	山内　△△	女
	(社員名)	(性別)

Eiyo	(0)	(1)	～	(19)	(20)	
(0)	15.9	11.2	～	71.0	194.0	(ｱｼﾞﾌﾗｲ)
≀	≀	≀	≀	≀	≀	≀
(499)	20.1	30.4	～	100.0	411.0	(ﾛｰﾙｷｬﾍﾞﾂ)
	(たんぱく質)	(脂質)	～	(ｺﾚｽﾃﾛｰﾙ)	(カロリー)	

２．第１図の入力データを読み，料理コードに対する栄養摂取量と摂取日数を配列 Kei に集計する。なお，Kei の行方向の添字は Syain と，列方向の添字は Eiyo と対応している。また，入力データは，日付，社員番号の昇順に記録されている。

配列

Kei	(0)	(1)	～	(19)	(20)	(21)
(0)			～			
(1)			～			
≀	≀	≀	≀	≀	≀	≀
(100)			～			
	(たんぱく質)	(脂質)	～	(ｺﾚｽﾃﾛｰﾙ)	(カロリー)	(摂取日数)

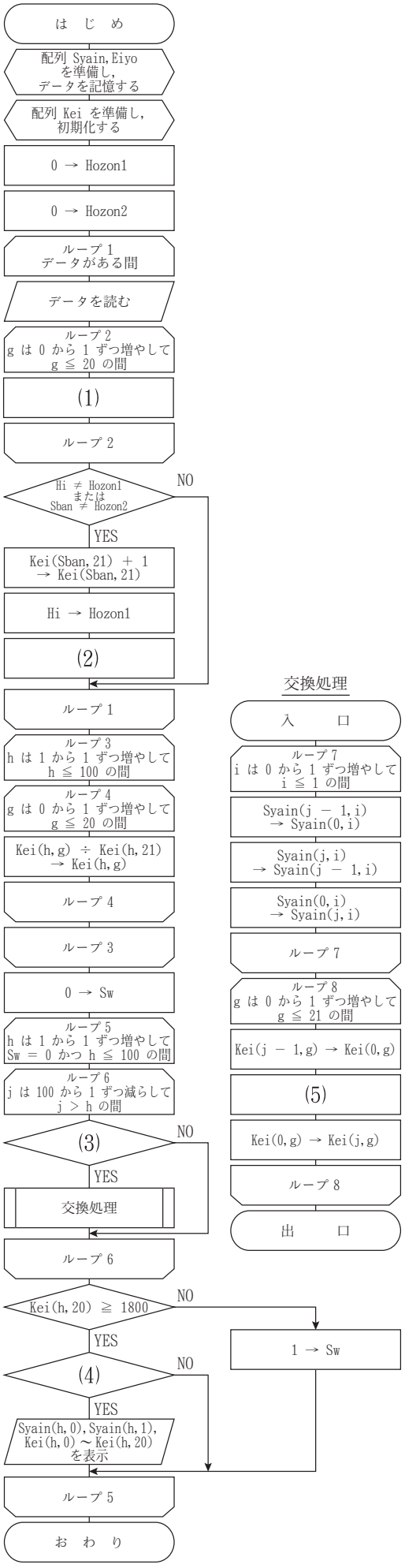
３．入力データが終了したら，配列 Kei の栄養摂取量を１日平均に換算し，配列 Syain とともにカロリーの降順に並べ替えながら，男性は2,300kcal以上，女性は1,800kcal以上の社員をディスプレイに表示する。

４．データにエラーはないものとする。

解答群

ア．Kei(j ＋ 1,20) ＜ Kei(j,20)
イ．Kei(Sban,g) ＋ Eiyo(Rcode,g) → Kei(Sban,g)
ウ．Syain(h,1) ＝ "男" かつ Kei(h,20) ≧ 2300
エ．Kei(j － 1,g) → Kei(j,g)
オ．Kei(j － 1,20) ＜ Kei(j,20)
カ．Syain(h,1) ＝ "女" または Kei(h,20) ≧ 2300
キ．Sban → Hozon2
ク．Kei(g,Sban) ＋ Eiyo(g,Rcode) → Kei(g,Sban)
ケ．Kei(j,g) → Kei(j － 1,g)
コ．Rcode → Hozon2

＜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

メニューデータと売上データを読み，弁当販売店の売上金額と順位を表示する。

入力データ

メニューデータ（ファイル名：menu.csv）

商品コード	商品名	単価
×××	×～×	××××

（第１図）

売上データ（ファイル名：sells.csv）

売上番号	商品コード	売上数量	販売種別
×××	×××	××	×

（第２図）

実行結果

(全体順位)	(分類別順位)	(商品コード)	(商品名)	(単価)	(売上数量計)	(売上金額)	(値引販売数量)
19	16	101	ステーキ弁当	760	5	3,800	0
≀	≀	≀	≀	≀	≀	≀	≀
9	8	135	煮魚弁当	460	12	5,120	4
33	10	201	マカロニサラダ	260	7	1,720	5
≀	≀	≀	≀	≀	≀	≀	≀
37	12	227	みそ汁	100	10	940	3

（第３図）

処理条件

１．第１図のメニューデータは50件であり，商品コードの昇順に記録されている。なお，商品コードは，100番台が弁当，200番台がサイドメニューであり，分類をあらわしている。

２．第１図のメニューデータを読み，弁当の場合はLunchMenuクラスを，サイドメニューの場合はSideMenuクラスをインスタンス化し，商品コード順に配列obj[0]～[49]に記憶する。なお，Menuクラスは，LunchMenuクラスとSideMenuクラスのスーパークラスであり，全ての商品分類の共通属性を管理している。

配列

obj	(0)	(1)	～	(48)	(49)
			～		

３．第２図の売上データを読み，商品コードをもとに配列objを探索し，売上数量と売上金額を集計する。なお，販売種別に0が記録されていたら通常単価での販売である。また，1が記録されていたら弁当の場合「ライス抜き（おかずのみ）の販売」であり，単価から100円を引き，サイドメニューの場合「弁当とのセット販売」であり，単価から20円を引いて売上金額を求める。

４．データを読み終えたあと，配列rankを利用して売上金額の降順に全体順位と分類別順位をつける。なお，売上金額が同じ場合は，同順位とする。また，配列objとrankは列方向の添字が対応している。

配列

rank	(0)	(1)	～	(48)	(49)	
(0)			～			(全体順位)
(1)			～			(分類別順位)

５．最後に，第３図のように全体順位から値引販売数量までをディスプレイに表示する。

＜Ｊａｖａプログラム＞

```
//クラスMenu
public class Menu {
    private int code;
    private String name;
    private int price;
    private int total;
    private int sales;
    public Menu(int code, String name, int price) {
        this.code = code;
        this.name = name;
        this.price = price;
        total = 0;
        sales = 0;
    }
    public int getCode() {
        return code;
    }
    public int getPrice() {
        return price;
    }
    public int getTotal() {
        return total;
    }
    public int getSales() {
        return sales;
    }
    public void setTotal(int total) {
        this.total = total;
    }
    public void setSales(int sales) {
        this.sales = sales;
    }
    public void calcSales(int quantity, int kind) {
        if(kind == 0) {
            [      (1)      ];
            sales += price * quantity;
        }
    }
    public void outList() {
        System.out.printf("¥t%3d ¥t%-10s ¥t%3d ¥t%2d ¥t%,5d ",
            code, name, price, total, sales);
    }
}
```

```
//クラスLunchMenu
public class LunchMenu extends Menu {
    private int noriceSell;
    public LunchMenu(int code, String name, int price) {
        super(code, name, price);
        noriceSell = 0;
    }
    public void calcSales(int quantity, int kind) {
        if(kind == 1) {
            setTotal(getTotal() + quantity);
            setSales(getSales() + (getPrice() - 100) * quantity);
            noriceSell += quantity;
        } else {
            super.calcSales(quantity, kind);
        }
    }
    public void outList() {
        super.outList();
        System.out.printf("¥t%2d ¥n", noriceSell);
    }
}

//クラスSideMenu
public class SideMenu extends Menu {
    private int combiSell;
    public SideMenu(int code, String name, int price) {
        super(code, name, price);
        combiSell = 0;
    }
    public void calcSales(int quantity, int kind) {
        if(kind == 1) {
            setTotal(getTotal() + quantity);
            setSales(getSales() + (getPrice() - 20) * quantity);
            combiSell += quantity;
        } else {
            super.calcSales(quantity, kind);
        }
    }
    public void outList() {
        super.outList();
        System.out.printf("¥t%2d ¥n", combiSell);
    }
}
```

［１級　Ｊａｖａ選択者のための問題］


```
//クラスSellAnalyst
import java.io.BufferedReader;
import java.io.FileReader;

public class SellAnalyst {
    private static final int SIZE = 50;
    private static final int SCOPE0 = 0;
    private static final int SCOPE1 = 1;
    private static Menu[] obj = new Menu[SIZE];
    private static int[][] rank = new int[2][SIZE];
    private static int idx;
    private static void addRank(int start, int end, int scope) {
        for(int [          (2)          ]; p++) {
            rank[scope][p] = 1;
        }
        for(int p = start; p < end; p++) {
            for(int q = p + 1; q <= end; q++) {
                if(obj[p].getSales() < obj[q].getSales()) {
                    [        解答不要        ];
                } else if(obj[p].getSales() > obj[q].getSales()) {
                    [          (3)           ];
                }
            }
        }
    }
    public static void main(String[] args) {
        try {
            BufferedReader fileIn1 = new BufferedReader(new FileReader("menu.csv"));
            String line;
            int i = 0;
            while((line = fileIn1.readLine()) != null) {
                String[] str = line.split(",");
                int code = Integer.parseInt(str[0]);
                String name = str[1];
                int price = Integer.parseInt(str[2]);
                int type = code / 100;
                switch(type) {
                  case 1:
                    obj[i] = new LunchMenu(code, name, price);
                    idx = i;
                    break;
                  case 2:
                    obj[i] = new SideMenu(code, name, price);
                    break;
                }
                i += 1;
            }
            fileIn1.close();
            BufferedReader fileIn2 = new BufferedReader(new FileReader("sells.csv"));
            while((line = fileIn2.readLine()) != null) {
                String[] str = line.split(",");
                int code = Integer.parseInt(str[1]);
                int quantity = Integer.parseInt(str[2]);
                int kind = Integer.parseInt(str[3]);
                int lower = 0;
                int upper = SIZE - 1;
                int m = (lower + upper) / 2;
                while(obj[m].getCode() != code) {
                    if([          (4)          ]) {
                        lower = m + 1;
                    } else {
                        upper = m - 1;
                    }
                    m = (lower + upper) / 2;
                }
                [                (5)                ];
            }
            fileIn2.close();
        } catch(Exception e) {
            System.out.println("エラーが発生しました" + e);
        }
        addRank(0, SIZE - 1, SCOPE0);
        addRank(0, idx, SCOPE1);
        addRank(idx + 1, SIZE - 1, SCOPE1);
        for(int n = 0; n < SIZE; n++) {
            System.out.printf("¥t%2d ¥t%2d ", rank[SCOPE0][n], rank[SCOPE1][n]);
            obj[n].outList();
        }
    }
}
```

【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

メニューデータと売上データを読み，分類別に順位を求め表示する。

入力データ

メニューデータ（ファイル名：menu.csv）

メニューコード	メニュー名	単価
×××	×～×	××××

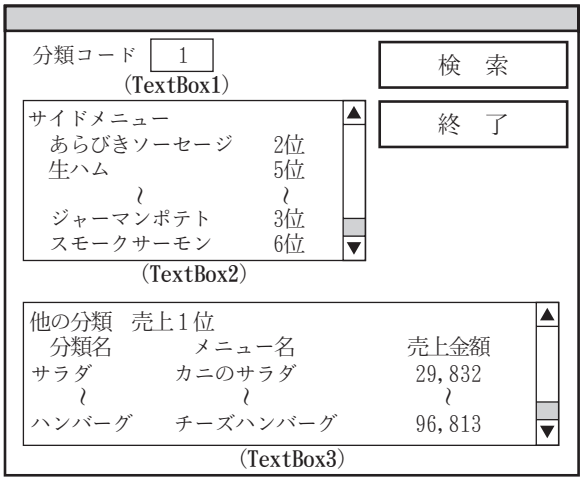
（第１図）

売上データ（ファイル名：uriage.csv）

伝票番号	メニューコード	売上数量	クーポン
×××	×××	××	×

（第２図）

ユーザーフォーム・実行結果



（第３図）

処理条件

１．第１図のメニューデータはメニューコードの昇順に記録されている。なお，メニューは60種類あり，次のように分類コードと商品コードで構成されている。また，分類コードは１～７の７種類である。

例　１２５　→　１（分類コード）　２５（商品コード）

２．第２図の売上データのクーポンは，０（クーポン使用なし）または，１（クーポン使用あり）が記録されており，１の場合は単価を10%引きする。なお，10円未満は切り捨てる。

３．ユーザーフォーム初期化時に，次の処理を行う。

・　配列 Bunrui に分類名を記憶する。

配列

	(0)	(1)	～	(7)
Bunrui		サイドメニュー	～	ハンバーグ

・　第１図のメニューデータを読み，配列 Code にメニューコード，配列 Menu にメニュー名，配列 Tanka に単価を記憶する。なお，配列 Code，Menu，Tanka，Syu の行方向，Kin，Jun の各添字は対応している。

・　第２図の売上データを読み，配列 Code を探索し，クーポン使用なしとクーポン使用ありに分けて，売上数量を配列 Syu に集計する。なお，Syu の列方向の添字はクーポンと対応している。

・　データを読み終えたあと，配列 Kin に各メニューの売上金額を求める。

・　配列 Jun を利用し，配列 Code に記憶されている分類コードをグループキーとし，分類コード別に売上金額の降順に順位をつける。

配列

	Code	Menu	Tanka	Syu (0)	Syu (1)	Kin	Jun
(0)	0						
(1)	101	あらびきソーセージ	310				
⌇	⌇	⌇	⌇	⌇	⌇	⌇	⌇
(60)	712	和牛ハンバーグ	730				
	(メニューコード)	(メニュー名)	(単価)	(売上数量計)		(売上金額)	(順位)

４．第３図のように TextBox1 に分類コードを入力し，「検索」ボタンをクリックすると，次の処理を行う。

・　分類コードをもとに配列 Code を探索し，TextBox2 に分類名とその分類に属するメニュー名，順位を表示する。

・　入力された分類コード以外の各分類の中で，順位が１位のメニューの分類名，メニュー名，売上金額を TextBox3 に表示する。

［１級　マクロ言語選択者のための問題］

＜マクロ言語プログラム＞

```
Option Explicit
Dim Bunrui(7) As String, Code(60) As Long, Menu(60) As String, Tanka(60) As Long, Syu(60, 1) As Long
Dim Kin(60) As Long, Jun(60) As Long
```

```
Private Sub UserForm_Initialize()
    Dim i As Long, Ban As Long, Cd As Long, Su As Long, Kupon As Long, Ka As Long, Jo As Long, m As Long, Hozon As Long
    Dim Owari As Long, s As Long, t As Long
    Bunrui(1) = "サイドメニュー": Bunrui(2) = "サラダ": ～ Bunrui(6) = "パスタ": Bunrui(7) = "ハンバーグ"
    Open ThisWorkbook.Path & "¥menu.csv" For Input As #1
    Code(0) = 0
    For i = 1 To 60
        Input #1, Code(i), Menu(i), Tanka(i)
        Syu(i, 0) = 0
        Syu(i, 1) = 0
        (1)
    Next i
    Close #1
    Open ThisWorkbook.Path & "¥uriage.csv" For Input As #2
    Do While Not EOF(2)
        Input #2, Ban, Cd, Su, Kupon
        Ka = 1
        Jo = 60
        m = Int((Ka + Jo) / 2)
        Do While Code(m) <> Cd
            If (2) Then
                Ka = m + 1
            Else
                Jo = m - 1
            End If
            m = Int((Ka + Jo) / 2)
        Loop
        (3)
    Loop
    Close #2
    For (4)
        Kin(i) = Syu(i, 0) * Tanka(i)
        Kin(i) = Kin(i) + Syu(i, 1) * Int((Tanka(i) * 0.9) / 10) * 10
    Next i
    i = 60
    Do While i <> 0
        Hozon = Int(Code(i) / 100)
        Owari = i
        Do While Int(Code(i) / 100) = Hozon
            i = i - 1
        Loop
        s = i + 1
        Do While s < Owari
            t = s + 1
            Do While t <= Owari
                If Kin(s) < Kin(t) Then
                    解答不要
                Else
                    If Kin(s) > Kin(t) Then
                        (5)
                    End If
                End If
                t = t + 1
            Loop
            s = s + 1
        Loop
    Loop
End Sub
```

```
Private Sub 検索_Click()
    Dim Kensaku As Long, i As Long, Hozon As Long
    Kensaku = Val(TextBox1.Text)
    TextBox2.Text = Bunrui(Kensaku) & Chr(13) & Chr(10)
    TextBox3.Text = "他の分類　売上１位" & Chr(13) & Chr(10) & "  分類名       メニュー名           売上金額" & Chr(13) & Chr(10)
    i = 1
    Do While i <= 60
        Hozon = Int(Code(i) / 100)
        If Hozon = Kensaku Then
            TextBox2.Text = TextBox2.Text & Menu(i) & "    " & Format(Jun(i), "#0") & "位" & Chr(13) & Chr(10)
        Else
            If Jun(i) = 1 Then
                TextBox3.Text = TextBox3.Text & Bunrui(Hozon) & "      " & Menu(i) & "       " & _
                    Format(Kin(i), "###,##0") & Chr(13) & Chr(10)
            End If
        End If
        i = i + 1
    Loop
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

旅行先ファイルと調査ファイルを読み，旅行先ごとのアンケート集計結果を出力する。

入力データ

旅行先ファイル
（ファイル名：RYOKO-F，レコード名：RYOKO-R）

旅コード (R-CODE)	旅行先 (R-NAME)
×××	×～×

（第１図）

調査ファイル
（ファイル名：TYOSA-F，レコード名：TYOSA-R）

旅行者番号 (T-NO)	旅コード (T-CODE)	評価点 (T-TEN)
×××	×××	×××

（第２図）

実行結果

プリンタ出力する地方コードを 1～8 で入力してください
1

（各地方の１位（参考データ））

（旅コード）	（旅行先）	（平均点）	（順位）	（地方名）
203	□～□	80.6	7	東　北
386	◆～◆	77.2	23	関　東
≀	≀	≀	≀	≀

（第３図）

（ファイル名：OUT-F，レコード名：OUT-R）

（旅行先アンケート結果）

（旅コード）	（旅行先）	（平均点）	（順位）
101	◎～◎	77.0	24
113	△～△	79.6	10
≀	≀	≀	≀

（第４図）

処理条件

１．第１図の旅行先ファイルは，旅コードの昇順に記録されている。なお，旅コードは100種類あり，次のように地方コードと場所コードで構成されている。また，地方コードは１～８の８種類である。

例　３０８　→　3（地方コード）　０８（場所コード）

２．旅行先ファイルを読み，テーブル DATA-TBL に旅コードと旅行先を記憶する。なお，テーブル TIHO-TBL にはあらかじめ地方名が記憶されており，地方コードと添字で対応している。

テーブル　DATA-TBL

	D-CODE	D-NAME	D-TEN	D-KEN	D-HEI	D-JUN
(1)	101	◎～◎				
(2)	113	△～△				
≀	≀	≀	≀	≀	≀	≀
(100)	880	★～★				
	（旅コード）	（旅行先）	（評価点計）	（件数）	（平均点）	（順位）

テーブル　TIHO-TBL

	T-TIHO
(1)	北海道
≀	≀
(8)	九　州

３．第２図の調査ファイルを読み，次の処理を行う。

- 旅コードをもとにテーブル DATA-TBL を探索し，D-TEN に評価点を，D-KEN に件数をそれぞれ集計する。

４．ファイルを読み終えたあと，次の処理を行う。

- テーブル DATA-TBL の評価点計と件数をもとに，D-HEI に旅行先ごとの平均点を求める。
- 平均点の降順に順位をつける。なお，同点の場合は同順位とする。
- 第３図のようにキーボードからプリンタに出力する地方コードを入力し，該当する地方の旅コードから順位までを第４図のように印字する。また，入力した地方コードに該当しない地方については，その地方の中で１位の旅コードから地方名までをディスプレイに表示する。ただし，評価点計が０点の旅行先は存在しない。

＜ＣＯＢＯＬプログラム＞

```
WORKING-STORAGE SECTION.
01  END-FLG            PIC X(01) VALUE LOW-VALUE.
01  TIHO-CODE          PIC 9(01).
      ≀
01  HOZON              PIC 9(01).
01  JUN                PIC 9(03).
01  DATA-TBL.
    02  D-DATA OCCURS 100.
        03  D-CODE.
            04  D-TC   PIC 9(01).
            04  D-RC   PIC 9(02).
        03  D-NAME     PIC X(20).
        03  D-TEN      PIC 9(05).
        03  D-KEN      PIC 9(03).
        03  D-HEI      PIC 9(03)V9(01).
        03  D-JUN      PIC 9(03).
01  TIHO-DATA.
    02                  PIC X(48) VALUE "北海道東　北関　東中　部近　畿中　国四　国九　州".
01  TIHO-TBL REDEFINES TIHO-DATA.
    02  T-TIHO         PIC X(06) OCCURS 8.
      ≀
```

```
PROCEDURE DIVISION.
P1. OPEN INPUT RYOKO-F TYOSA-F OUTPUT OUT-F
    PERFORM VARYING N FROM 1 BY 1 UNTIL N > 100
        READ RYOKO-F
          NOT AT END
            MOVE R-CODE TO D-CODE(N)
            MOVE R-NAME TO D-NAME(N)
            MOVE 0      TO D-TEN(N) D-KEN(N)
        END-READ
    END-PERFORM
    PERFORM UNTIL END-FLG = HIGH-VALUE
        READ TYOSA-F
          AT END
            MOVE HIGH-VALUE TO END-FLG
          NOT AT END
            MOVE   1 TO KA
            MOVE 100 TO JO
            COMPUTE M = (KA + JO) / 2
            PERFORM UNTIL D-CODE(M) = T-CODE
                IF [        (1)        ]
                  THEN
                    COMPUTE KA = M + 1
                  ELSE
                    COMPUTE JO = M - 1
                END-IF
                COMPUTE M = (KA + JO) / 2
            END-PERFORM
            [                  (2)                  ]
            COMPUTE D-KEN(M) = D-KEN(M) + 1
        END-READ
    END-PERFORM
    PERFORM VARYING N FROM 1 BY 1 UNTIL N > 100
        COMPUTE D-HEI(N) = D-TEN(N) / D-KEN(N)
        [        (3)        ]
    END-PERFORM
    PERFORM VARYING N FROM 1 BY 1 UNTIL N > 99
        COMPUTE P = N + 1
        PERFORM VARYING Q FROM P BY 1 UNTIL Q > 100
            EVALUATE TRUE
              WHEN D-HEI(N) < D-HEI(Q)
                [                解答不要                ]
              WHEN D-HEI(N) > D-HEI(Q)
                [                  (4)                  ]
            END-EVALUATE
        END-PERFORM
    END-PERFORM
    DISPLAY "プリンタ出力する地方コードを 1～8 で入力してください"
    ACCEPT TIHO-CODE
    MOVE 1 TO R
    PERFORM UNTIL R > 100
        MOVE D-TC(R) TO HOZON
        [        (5)        ]
        PERFORM UNTIL R > 100 OR D-TC(R) NOT = HOZON
            COMPUTE R = R + 1
        END-PERFORM
        COMPUTE OWARI = R - 1
        IF HOZON = TIHO-CODE
          THEN
            PERFORM VARYING S FROM HAJIME BY 1 UNTIL S > OWARI
                PERFORM P2
                WRITE OUT-R FROM MEISAI AFTER 1
            END-PERFORM
          ELSE
            PERFORM VARYING S FROM HAJIME BY 1 UNTIL S > OWARI
                MOVE 1 TO JUN
                PERFORM VARYING T FROM HAJIME BY 1 UNTIL JUN > 1 OR T > OWARI
                    IF D-HEI(S) < D-HEI(T)
                      THEN
                        COMPUTE JUN = JUN + 1
                    END-IF
                END-PERFORM
                IF JUN = 1
                  THEN
                    PERFORM P2
                    DISPLAY MEISAI "    " T-TIHO(HOZON)
                END-IF
            END-PERFORM
        END-IF
    END-PERFORM
    CLOSE RYOKO-F TYOSA-F OUT-F
    STOP RUN.
P2. MOVE D-CODE(S) TO M-CODE
    MOVE D-NAME(S) TO M-NAME
    MOVE D-HEI(S)  TO M-HEI
    MOVE D-JUN(S)  TO M-JUN.
```

（平成26年１月19日実施）

主催　公益財団法人 全国商業高等学校協会

平成25年度（第50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

……〔Ｊａｖａ〕・〔マクロ言語〕・〔ＣＯＢＯＬ〕……

【７】

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

試験場校名	受験番号	選択言語		
		Ｊａｖａ	マクロ言語	COBOL

小　計

合　計

選択言語を（　　）で囲むこと

（平成26年１月19日実施）

主催　公益財団法人 全国商業高等学校協会

平成25年度（第50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
オ	サ	ア	ケ	カ

各２点 計10点

【2】

1	2	3	4	5
コ	エ	ク	キ	イ

各２点 計10点

【3】

1	2	3	4	5
ウ	イ	ウ	イ	ア

各２点 計10点

小　計
30

【4】

(1)	(2)	(3)	(4)	(5)
カ	ウ	オ	エ	ア

各３点 計15点

【5】

(1)	(2)	(3)	(4)	(5)
エ	ア	コ	ク	ウ

各３点 計15点

【6】

(1)	(2)	(3)	(4)	(5)
イ	キ	オ	カ	ケ

各３点 計15点

小　計
45

……〔Ｊａｖａ〕・〔マクロ言語〕・〔ＣＯＢＯＬ〕……

〔Ｊａｖａ〕　㊟ ＝，演算子の前後の空白は問わない。

【7】

(1)	ｔｏｔａｌ ＋＝ ｑｕａｎｔｉｔｙ
(2)	ｐ ＝ ｓｔａｒｔ； ｐ ＜＝ ｅｎｄ
(3)	ｒａｎｋ［ｓｃｏｐｅ］［ｑ］ ＋＝ １
(4)	ｏｂｊ［ｍ］．ｇｅｔＣｏｄｅ（） ＜ ｃｏｄｅ
(5)	ｏｂｊ［ｍ］．ｃａｌｃＳａｌｅｓ（ｑｕａｎｔｉｔｙ， ｋｉｎｄ）

〔マクロ言語〕 ㊟ 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)	Ｊｕｎ（ｉ） ＝ １
(2)	Ｃｏｄｅ（ｍ） ＜ Ｃｄ
(3)	Ｓｙｕ（ｍ， Ｋｕｐｏｎ） ＝ Ｓｙｕ（ｍ， Ｋｕｐｏｎ） ＋ Ｓｕ
(4)	ｉ ＝ １ Ｔｏ ６０
(5)	Ｊｕｎ（ｔ） ＝ Ｊｕｎ（ｔ） ＋ １

〔ＣＯＢＯＬ〕

【7】

(1)	Ｄ－ＣＯＤＥ（Ｍ） ＜ Ｔ－ＣＯＤＥ
(2)	ＣＯＭＰＵＴＥ Ｄ－ＴＥＮ（Ｍ） ＝ Ｄ－ＴＥＮ（Ｍ） ＋ Ｔ－ＴＥＮ
(3)	ＭＯＶＥ １ ＴＯ Ｄ－ＪＵＮ（Ｎ）
(4)	ＣＯＭＰＵＴＥ Ｄ－ＪＵＮ（Ｑ） ＝ Ｄ－ＪＵＮ（Ｑ） ＋ １
(5)	ＭＯＶＥ Ｒ ＴＯ ＨＡＪＩＭＥ

各５点　計25点

試験場校名	受験番号	選択言語		
		Ｊａｖａ	マクロ言語	COBOL

小　計
25

合　計
100

選択言語を（　　）で囲むこと