

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2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50分です。

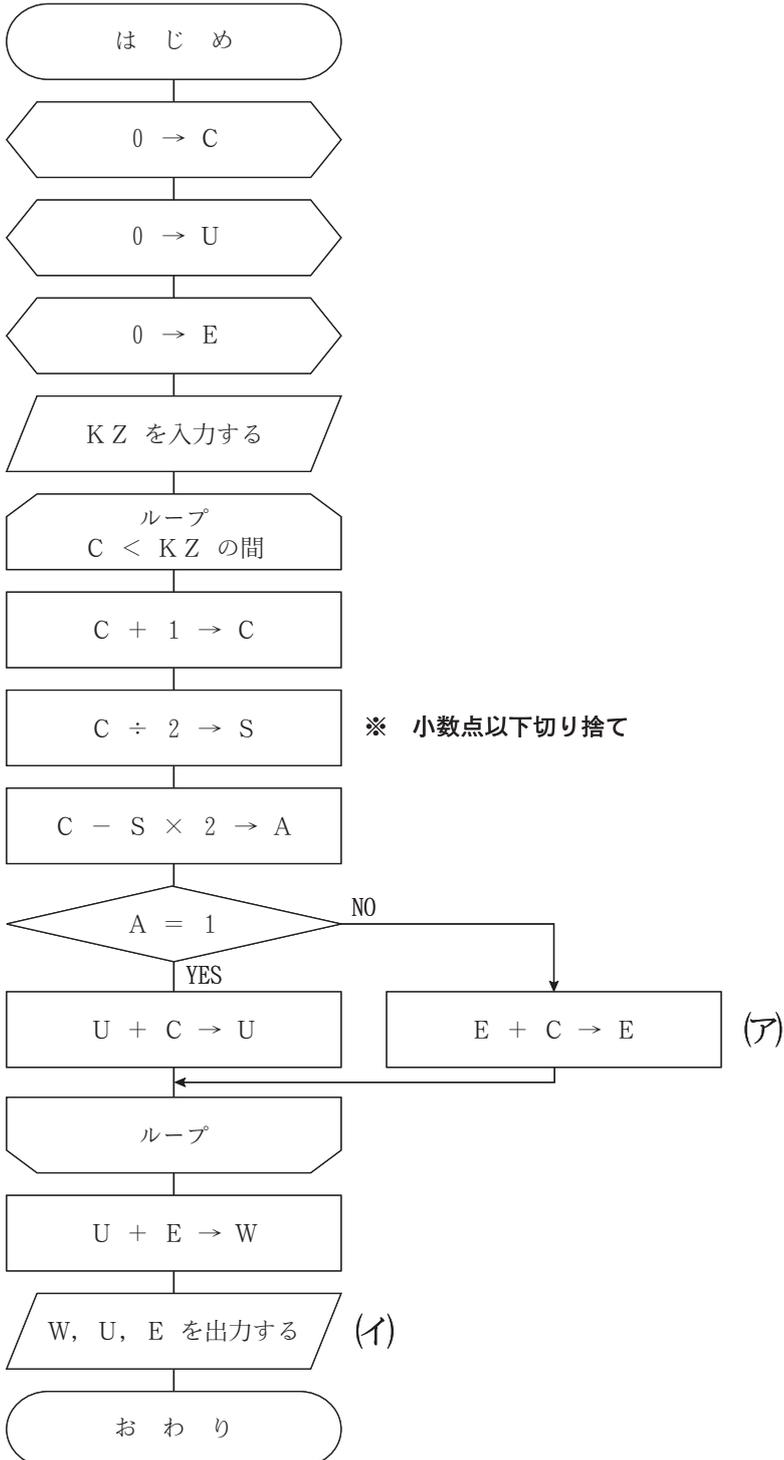
主催 公益財団法人 全国商業高等学校協会



【 4 】 流れ図にしたがって処理するとき、(1)～(5)を答えなさい。

- (1) K Z の値が 3 のとき、(ア)の処理を何回実行するか答えなさい。
- (2) K Z の値が 3 のとき、(イ)で出力される W の値を答えなさい。
- (3) K Z の値が 8 のとき、(イ)で出力される U の値を答えなさい。
- (4) K Z の値が 8 のとき、(イ)で出力される E の値を答えなさい。
- (5) 流れ図の処理について説明した文のうち、正しいものはどれかア、イ、ウの中から選び、記号で答えなさい。
  - ア. 出力される E の値は、1～K Z の和である。
  - イ. 出力される E の値は、1～K Z の奇数の和である。
  - ウ. 出力される E の値は、1～K Z の偶数の和である。

<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

都道府県の投票データを読み、投票率一覧をディスプレイに表示する。

入力データ

|                |                 |                 |
|----------------|-----------------|-----------------|
| 都道府県名<br>(Mei) | 有権者数<br>(Yuken) | 投票者数<br>(Tohyo) |
| ×～×            | ××××××××        | ××××××××        |

(第 1 図)

実行結果

| (投票率一覧) |           |           |          |       |
|---------|-----------|-----------|----------|-------|
| (都道府県名) | (有権者数)    | (投票者数)    | (投票率(%)) | (関心度) |
| ●～●     | 1,627,344 | 972,784   | 59.8     | △     |
| ▲～▲     | 1,988,332 | 1,336,655 | 67.2     | ◎     |
| ◆～◆     | 1,695,775 | 1,089,663 | 64.3     | ○     |
| }       | }         | }         | }        | }     |
| ■～■     | 1,504,275 | 957,517   | 63.7     | ○     |
| ▼～▼     | 7,186,849 | 4,096,955 | 57.0     | △     |
| ★～★     | 605,921   | 485,542   | 80.1     | ◎     |
|         |           | (全国)      | 60.9     |       |
|         |           | (最大)      | 80.1     |       |

(第 2 図)

処理条件

- 第 1 図の入力データを読み、投票率(%)を次の計算式で求め、第 2 図のように表示する。なお、関心度は、投票率(%)が65%以上の場合「◎」を、60%以上65%未満の場合「○」を、60%未満の場合「△」を表示する。

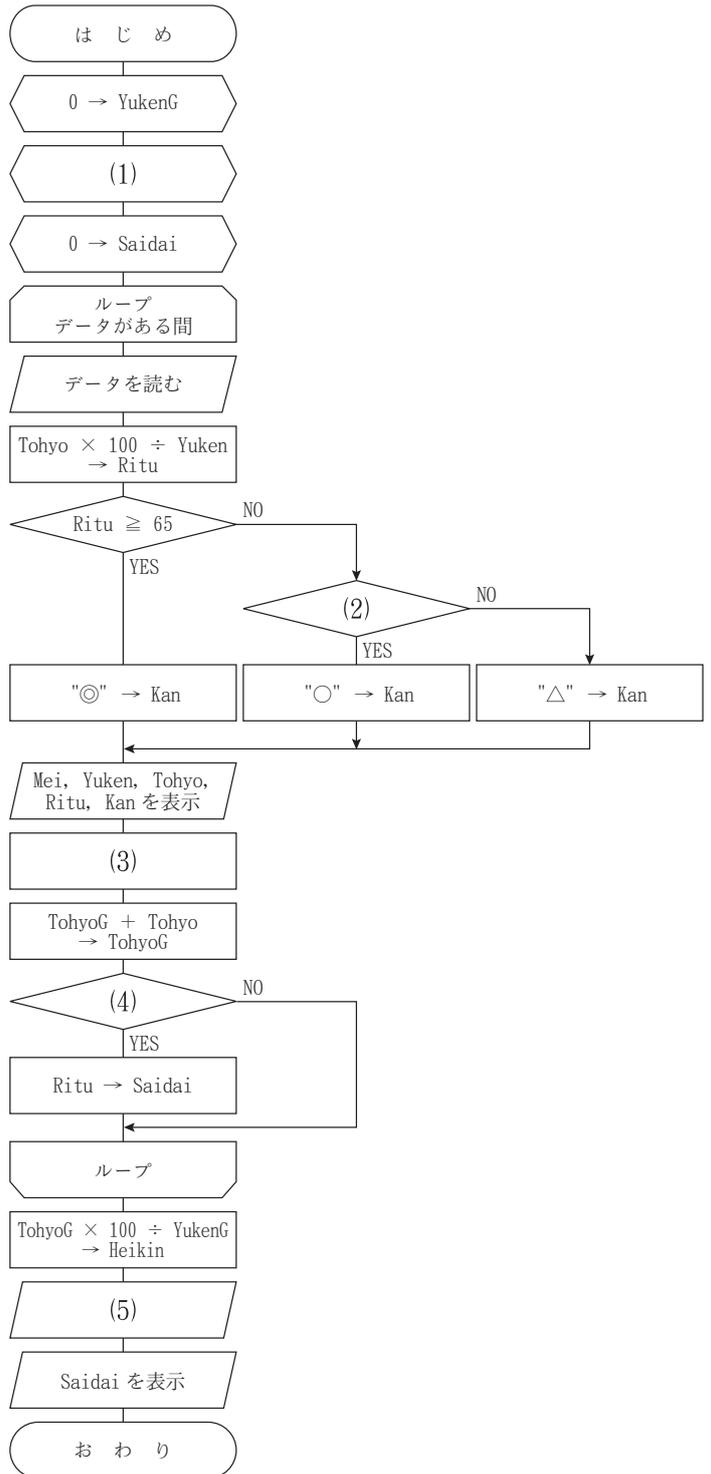
$$\text{投票率(\%)} = \text{投票者数} \times 100 \div \text{有権者数}$$

- 入力データが終了したら、第 2 図のように全国の投票率と投票率の最大を表示する。なお、最大は同じ投票率があった場合、先のデータを優先する。
- データにエラーはないものとする。

解答群

- ア. Ritu > Saidai
- イ. YukenG, TohyoG を表示
- ウ. YukenG + Tohyo → YukenG
- エ. Ritu < 65
- オ. YukenG + Yuken → YukenG
- カ. 0 → TohyoG
- キ. Ritu < Saidai
- ク. Ritu ≥ 60
- ケ. Heikin を表示
- コ. 1 → TohyoG

<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

市民講座の申込データを読み、申込一覧表と講座別申込状況をディスプレイに表示する。

入力データ

|                       |                        |                        |
|-----------------------|------------------------|------------------------|
| 受付番号<br>(Uban)<br>××× | 名前<br>(Namee)<br>××～×× | 申込コード<br>(Mcod)<br>××× |
|-----------------------|------------------------|------------------------|

(第 1 図)

実行結果

|            |         |         |            |
|------------|---------|---------|------------|
| (受付番号)     | (申込一覧表) |         | (講座名)      |
| 001        | 香川 太郎   | はじめての陶芸 |            |
| 641        | 花田 哲夫   | ドイツ料理入門 |            |
| (講座名)      | (市内)    | (市外)    | (申込数) (備考) |
| パソコン初級     | 18      | 6       | 24         |
| ドイツ料理入門    | 26      | 15      | 41 抽選      |
| (合計)       | 477     | 164     | 641        |
| (市内の割合(%)) | 74      |         |            |

(第 2 図)

処理条件

- 第 1 図の申込コードは、次の例のように構成されている。  
なお、区分は 1 が市内、2 が市外であり、講座番号は 1～20 である。

例 1 1 2 →  $\frac{1}{\text{区分}}$   $\frac{12}{\text{講座番号}}$

- 配列 Hkmei に講座名、配列 Hktei に定員を記憶する。  
なお、各配列の添字は講座番号と対応している。

配列

|       |     |        |   |         |
|-------|-----|--------|---|---------|
| Hkmei | (0) | (1)    | ～ | (20)    |
|       |     | パソコン初級 | ～ | ドイツ料理入門 |
| Hktei | (0) | (1)    | ～ | (20)    |
|       |     | 35     | ～ | 15      |

- 第 1 図のデータを読み、次の処理を行う。
  - 受付番号から講座名までを第 2 図のように表示する。
  - 区分が市内の場合は配列 Hnai に、市外の場合は配列 Hgai に申込数を集計する。なお、各配列の添字は講座番号と対応している。

配列

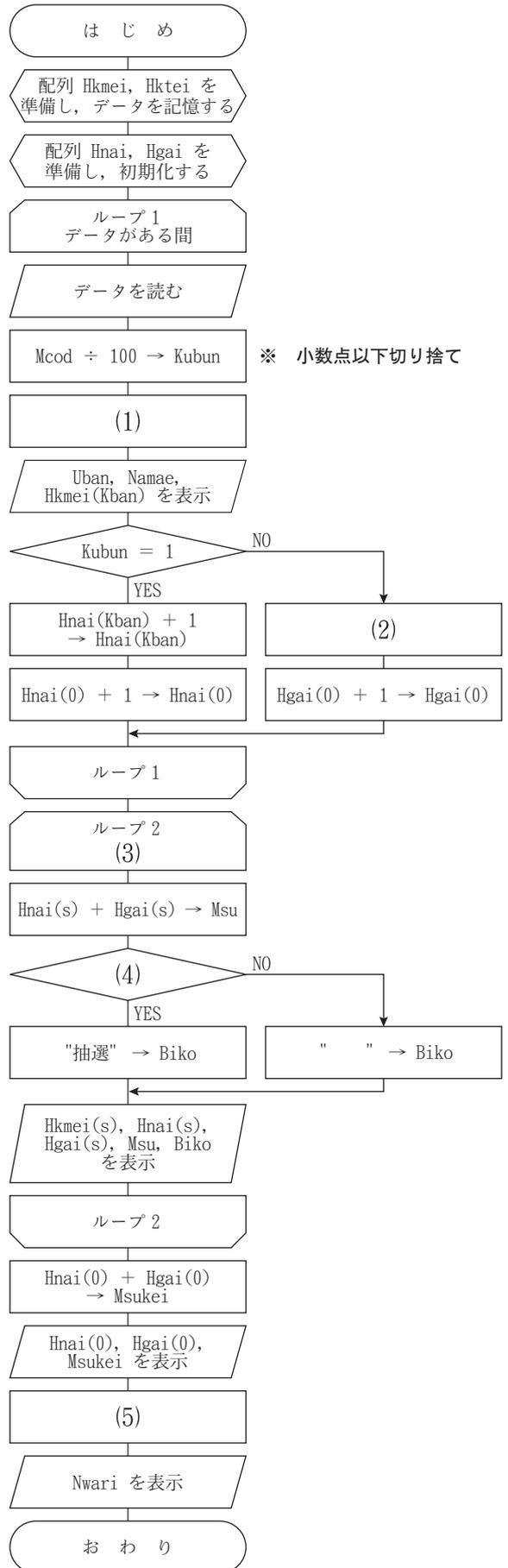
|      |     |        |   |      |
|------|-----|--------|---|------|
| Hnai | (0) | (1)    | ～ | (20) |
|      |     | (市内合計) | ～ |      |
| Hgai | (0) | (1)    | ～ | (20) |
|      |     | (市外合計) | ～ |      |

- 入力データが終了したら、次の処理を行う。
  - 講座別の申込数を求め、講座別申込状況を第 2 図のように表示する。なお、講座別の申込数が定員を超えた場合、備考に「抽選」を表示する。
  - 申込数合計を求め、市内合計、市外合計とともに第 2 図のように表示する。
  - 最後に、市内の割合(%)を次の計算式で求め、第 2 図のように表示する。  
 $\text{市内の割合(\%)} = \text{市内合計} \times 100 \div \text{申込数合計}$
- データにエラーはないものとする。

解答群

- ア. s は 0 から 1 ずつ増やして  $s < 20$  の間
- イ.  $\text{Hgai}(\text{Kban}) + 1 \rightarrow \text{Hgai}(\text{Kban})$
- ウ.  $\text{Hnai}(0) \times 100 \div \text{Msukei} \rightarrow \text{Nwari}$
- エ.  $\text{Mcod} - \text{Kubun} \times 100 \rightarrow \text{Kban}$
- オ.  $\text{Msu} \leq \text{Hktei}(s)$
- カ.  $\text{Hgai}(\text{Kubun}) + 1 \rightarrow \text{Hgai}(\text{Kubun})$
- キ.  $\text{Msu} > \text{Hktei}(s)$
- ク.  $\text{Mcod} \times 100 - \text{Kubun} \rightarrow \text{Kban}$
- ケ.  $\text{Hgai}(0) \times 100 \div \text{Msukei} \rightarrow \text{Nwari}$
- コ. s は 1 から 1 ずつ増やして  $s \leq 20$  の間

<流れ図>



【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

誕生日を入力し、12星座をディスプレイに表示する。

入力データ

誕生日  
××××× (第1図)

実行結果

誕生日の月日を入力してください  
誕生日：1221  
[星座は] いて座

誕生日：123  
[星座は] みずがめ座

誕生日：801  
[星座は] しし座

誕生日：0 (第2図)

処理条件

1. 配列 seizaBi に星座期間の最終日を、配列 seizaMei に星座名を記憶する。なお、各配列は添字で対応している。

配列

|          |                  |                   |   |                     |                     |
|----------|------------------|-------------------|---|---------------------|---------------------|
| seizaBi  | (0)              | (1)               | ~ | (11)                | (12)                |
|          | 119              | 218               | ~ | 1221                | 1231                |
|          | [1月1日~<br>1月19日] | [1月20日~<br>2月18日] | ~ | [11月22日~<br>12月21日] | [12月22日~<br>12月31日] |
| seizaMei | (0)              | (1)               | ~ | (11)                | (12)                |
|          | やぎ座              | みずがめ座             | ~ | いて座                 | やぎ座                 |

2. キーボードから第2図のように、誕生日の月日を入力する。なお、配列 seizaBi に記憶されている最終日および誕生日は、次の例のように構成されている。

例 307 → 3月7日

3. 誕生日をもとに配列 seizaBi を探索し、第2図のように表示する。

4. キーボードから誕生日に 0 が入力されたら処理を終了する。

<Javaプログラム>

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

public class Seiza {
    public static void main(String[] args) {
        int tanjoBi, i;
        int[] seizaBi = { 119, 218, ~ 1221, 1231 };
        String[] seizaMei = { "やぎ座", "みずがめ座", ~
            "いて座", "やぎ座" };
        Scanner sc = new Scanner(System.in);
        System.out.println("誕生日の月日を入力してください");
        System.out.print("誕生日：");
        (1) = sc.nextInt();
        while(tanjoBi > 0) {
            i = 0;
            while((2) < tanjoBi) {
                i = i + 1;
            }
            System.out.printf("[星座は] %-5s\n", seizaMei[i]);
            System.out.println("");
            System.out.print("誕生日：");
            tanjoBi = sc.nextInt();
        }
        sc.close();
    }
}
```

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

テーマパークの入場者数データを読み、アトラクション別とエリア別の入場者数をディスプレイに表示する。

入力データ

入場者数データ (ファイル名: nyujo.csv)

| 日  | アトラクションコード | エリアコード | 入場者数 |
|----|------------|--------|------|
| ×× | ××         | ×      | ×××× |

(第1図)

実行結果

| (アトラクション別入場者数) |         |         |
|----------------|---------|---------|
| (アトラクション名)     | (入場者数計) | (人気度)   |
| フォレストA         | 7,304   | ★★★★    |
| ネイチャー1         | 8,264   | ★★★★★   |
| }              | }       | }       |
| ファイヤーフラッシュ     | 3,962   | ★       |
| カヌーでゴー         | 7,416   | ★★★★    |
| (エリア別入場者数)     |         |         |
| (エリア名)         | (入場者数計) | (1日の平均) |
| 森と炎            | 47,160  | 6,737   |
| 海と水            | 56,712  | 8,101   |
| 不思議            | 40,057  | 5,722   |
| キッズ            | 15,531  | 2,218   |

(第2図)

処理条件

- 第1図のアトラクションコードは1~20, エリアコードは1~4であり, 1週間分のデータが記録されている。
- 配列 atorakuMei にアトラクション名を, 配列 eriaMei にエリア名を記憶する。なお, atorakuMei の添字はアトラクションコードと, eriaMei の添字はエリアコードと対応している。

配列

|            |     |        |     |        |     |
|------------|-----|--------|-----|--------|-----|
| atorakuMei | (0) | (1)    | ~   | (20)   |     |
|            |     | フォレストA | ~   | カヌーでゴー |     |
| eriaMei    | (0) | (1)    | (2) | (3)    | (4) |
|            |     | 森と炎    | 海と水 | 不思議    | キッズ |

- テーマパークのエリアを管理する Eria クラスをインスタンス化し, 配列 eriaJoho に記憶する。なお, eriaJoho の添字は, エリアコードと対応している。

配列

|          |     |     |     |     |     |
|----------|-----|-----|-----|-----|-----|
| eriaJoho | (0) | (1) | (2) | (3) | (4) |
|          |     |     |     |     |     |

- 第1図の入場者数データを読み, 入場者数を配列 atorakuKei に集計する。なお, atorakuKei の添字は, アトラクションコードと対応している。また, エリア別に入場者数を集計する。

配列

|            |     |     |   |      |
|------------|-----|-----|---|------|
| atorakuKei | (0) | (1) | ~ | (20) |
|            |     |     | ~ |      |

- データを読み終えたあと, アトラクション別入場者数を第2図のように表示する。なお, 人気度は, 2千人ごとに「★」を表示する。
- 最後に, エリアごとの1日の平均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。

$$1 \text{ 日の平均} = \text{入場者数計} \div 7$$

<Javaプログラム>

//クラスEria

```
public class Eria {
    public String eriaMei;
    public int nyujoKei = 0;
    public Eria(String eriaMei) {
        this.eriaMei = eriaMei;
    }
    public void gokei(int nyujo) {
        nyujoKei = nyujoKei + nyujo;
    }
    public void hyoji() {
        int heikin = (3);
        System.out.printf(" %-3s  %7d  %6d\n", eriaMei, nyujoKei, heikin);
    }
}
```

//クラスTemaPaku

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TemaPaku {
    public static void main(String[] args) throws IOException {
        String[] atorakuMei = { "", "フォレストA", "カヌーでゴー" };
        String[] eriaMei = { "", "森と炎", "海と水", "不思議", "キッズ" };
        Eria[] eriaJoho = new Eria[5];
        int[] atorakuKei = new int[21];
        for(int h = 1; h <= 4; h++) {
            eriaJoho[h] = new Eria(eriaMei[h]);
        }
        for(int i = 1; i <= 20; i++) {
            atorakuKei[i] = 0;
        }
        BufferedReader fileIn = new BufferedReader(new FileReader("nyujo.csv"));
        String line;
        while((line = fileIn.readLine()) != null) {
            String[] str = line.split(",");
            int hi = Integer.parseInt(str[0]);
            int atorakuCode = Integer.parseInt(str[1]);
            int eriaCode = Integer.parseInt(str[2]);
            int nyujo = Integer.parseInt(str[3]);
            atorakuKei[atorakuCode] = (4);
            eriaJoho[eriaCode].gokei(nyujo);
        }
        fileIn.close();
        for(int i = 1; i <= 20; i++) {
            System.out.printf("%-14s %6d  ", atorakuMei[i], atorakuKei[i]);
            int j = atorakuKei[i] / 2000;
            for(int k = 1; k <= j; k++) {
                System.out.print("★");
            }
            System.out.println("");
        }
        for(int h = 1; (5); h++) {
            eriaJoho[h].hyoji();
        }
    }
}
```

(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。

【 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

ユーザーフォームからレンタル自転車店の貸出データを入力し、車種別時間区分別貸出回数と貸出状況一覧をディスプレイに表示する。

ユーザーフォーム・実行結果

入力欄

|            |            |            |    |
|------------|------------|------------|----|
| 車種         | 開始時刻       | 利用時間       | 処理 |
| 2          | 17 時から     | 1 時間       |    |
| (TextBox1) | (TextBox2) | (TextBox3) | 終了 |

<車種別時間区分別貸出回数>

|         |        |        |        |       |          |
|---------|--------|--------|--------|-------|----------|
|         | ~ 2 時間 | ~ 4 時間 | ~ 8 時間 | ~12時間 |          |
| 電動補助自転車 | 2      | 3      | 2      | 2     | (Label1) |
| シティ自転車  | 3      | 2      | 2      | 0     | (Label2) |

<貸出状況一覧>

| 車種 | 開始時刻 | 利用時間 | 料金    |   |
|----|------|------|-------|---|
| 1  | 8    | 11   | 1,430 | ▲ |
| 1  | 8    | 10   | 1,430 |   |
| 1  | 9    | 3    | 490   |   |
| 2  | 9    | 5    | 740   |   |
| }  | }    | }    | }     |   |
| 1  | 15   | 3    | 490   |   |
| 2  | 15   | 3    | 380   |   |
| 2  | 16   | 2    | 200   |   |
| 2  | 17   | 1    | 200   | ▼ |

料金合計: 9,610 (TextBox5)

(第 1 図)

処理条件

1. TextBox1 に車種, TextBox2 に開始時刻, TextBox3 に利用時間を 1 時間単位で入力する。なお, 車種は 1 が電動補助自転車, 2 がシティ自転車の 2 種類であり, 利用時間は 12 時間以内である。
2. ユーザーフォーム初期化時に次の処理を行う。
  - ・ 配列 Ji に時間区分の上限を, 配列 Ryo に時間区分に対するシティ自転車の料金をそれぞれ記憶する。なお, 各配列は添字で対応している。

配列

|     |     |          |          |          |         |
|-----|-----|----------|----------|----------|---------|
| Ji  | (0) | (1)      | (2)      | (3)      | (4)     |
|     |     | 2        | 4        | 8        | 12      |
| Ryo | (0) | (1)      | (2)      | (3)      | (4)     |
|     |     | 200      | 380      | 740      | 1100    |
|     |     | (~ 2 時間) | (~ 4 時間) | (~ 8 時間) | (~12時間) |

3. 第 1 図のように入力欄に入力し, 「処理」ボタンをクリックすると次の処理を行う。
  - ・ 利用時間をもとに配列 Ji を探索し, 時間区分に対する料金を求める。なお, 電動補助自転車については, シティ自転車の料金の 1.3 倍であり, 10 円未満を切り捨てて計算する。
  - ・ 車種が電動補助自転車の場合は配列 DaiD に, シティ自転車の場合は配列 DaiS に, それぞれ時間区分ごとの貸出回数を集計する。なお, DaiD と DaiS の添字は, 配列 Ji, 配列 Ryo と対応している。

配列

|      |     |          |          |          |         |
|------|-----|----------|----------|----------|---------|
| DaiD | (0) | (1)      | (2)      | (3)      | (4)     |
|      |     |          |          |          |         |
| DaiS | (0) | (1)      | (2)      | (3)      | (4)     |
|      |     |          |          |          |         |
|      |     | (~ 2 時間) | (~ 4 時間) | (~ 8 時間) | (~12時間) |

- ・ 車種が電動補助自転車の場合は Label1 に, シティ自転車の場合は Label2 に, 現在までの貸出回数を表示する。
- ・ 車種から料金までを TextBox4 に表示する。
- ・ 料金合計を求め, TextBox5 に表示する。

## &lt;マクロ言語プログラム&gt;

Option Explicit

Dim Ji(4) As Long, Ryo(4) As Long, DaiD(4) As Long, DaiS(4) As Long, Ryokinkei As Long

Private Sub UserForm\_Initialize()

Dim i As Long

TextBox1.Text = "": TextBox2.Text = "": TextBox3.Text = "": TextBox4.Text = "": TextBox5.Text = ""

Label1.Caption = " 0 0 0 0"

Label2.Caption = " 0 0 0 0"

Ji(1) = 2: Ji(2) = 4: Ji(3) = 8: Ji(4) = 12

Ryo(1) = 200: Ryo(2) = 380: Ryo(3) = 740: Ryo(4) = 1100

For i = 1 To 4

DaiD(i) = 0

DaiS(i) = 0

Next i

Ryokinkei = 0

End Sub

Private Sub 処理\_Click()

Dim Syasyu As Long, Kaisi As Long, Riyo As Long, Soe As Long, Ryokin As Long, m As Long

Syasyu = Val(TextBox1.Text)

Kaisi = Val(TextBox2.Text)

Riyo = Val(TextBox3.Text)

Soe = 1

Do While  < Riyo

Soe = Soe + 1

Loop

Ryokin = Ryo(Soe)

If  Then

Ryokin = Int(Ryokin \* 1.3 / 10) \* 10

DaiD(Soe) = DaiD(Soe) + 1

Label1.Caption = ""

For m = 1 To 4

Label1.Caption = Label1.Caption & " " & Format(, "#0")

Next m

Else

DaiS(Soe) = 

Label2.Caption = ""

For m = 1 To 4

Label2.Caption = Label2.Caption &amp; " " &amp; Format(DaiS(m), "#0")

Next m

End If

TextBox4.Text = TextBox4.Text &amp; " " &amp; Format(Syasyu, "0") &amp; " " &amp; Format(Kaisi, "#0") &amp; " " &amp; \_

Format(Riyo, "#0") &amp; " " &amp; Format(Ryokin, "##0") &amp; Chr(13) &amp; Chr(10)

Ryokinkei = 

TextBox5.Text = Format(Ryokinkei, "##,##0")

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

紳士服店のキャンペーン中に配付した粗品配付ファイルを読み、粗品配付一覧と粗品別配付数を印字する。

入力データ

粗品配付ファイル (ファイル名: HAIFU-F, レコード名: HAIFU-R)

|                      |                             |
|----------------------|-----------------------------|
| 日付<br>(H-HI)<br>×××× | 会員番号<br>(H-BAN)<br>×××××××× |
|----------------------|-----------------------------|

(第1図)

実行結果

|                            |
|----------------------------|
| キャンペーン実施年を入力してください<br>2014 |
|----------------------------|

(第2図)

(ファイル名: OUT-F, レコード名: OUT-R)

| (粗品配付一覧) |               |       |         |       |
|----------|---------------|-------|---------|-------|
| (日付)     | (会員番号)        | (粗品名) | (新規入会)  |       |
| 1212     | 20100131      | ペン    |         |       |
| 1212     | 20140305      | 定規    |         | ○     |
| 1212     | 20071278      | メモ帳   |         |       |
| }        | }             | }     |         | }     |
| 1221     | 20010089      | タオル   |         |       |
| 1221     | 20130562      | 定規    |         |       |
| 1221     | 20041064      | マグカップ |         |       |
|          | (粗品配付件数)      |       | 255     |       |
| (粗品別配付数) |               |       |         |       |
| (定規)     | (ペン)          | (メモ帳) | (マグカップ) | (タオル) |
| 60       | 48            | 56    | 19      | 72    |
|          | (マグカップの割合(%)) |       | 7.5     |       |

(第3図)

処理条件

1. 第1図の会員番号は、次の例のように構成されている。

例 20120156 → 2012 0156  
                  入会年      連番

2. テーブル KEIKA-T には入会してからの経過年数の上限が、テーブル SMEI-T には粗品名があらかじめ記憶されている。なお、各テーブルは添字で対応している。

テーブル KEIKA-T

|       |       |       |       |       |        |
|-------|-------|-------|-------|-------|--------|
| KEIKA | (1)   | (2)   | (3)   | (4)   | (5)    |
|       | 3     | 6     | 9     | 10    | 99     |
|       | (~3年) | (~6年) | (~9年) | (10年) | (11年~) |

テーブル SMEI-T

|      |     |     |     |       |     |
|------|-----|-----|-----|-------|-----|
| SMEI | (1) | (2) | (3) | (4)   | (5) |
|      | 定規  | ペン  | メモ帳 | マグカップ | タオル |

3. キーボードから第2図のように、キャンペーン実施年を入力する。

4. 第1図のファイルを読み、次の処理を行う。

- 経過年数を次の計算式で求める。

$$\text{経過年数} = \text{キャンペーン実施年} - \text{入会年}$$

- 経過年数をもとにテーブル KEIKA-T を探索し、粗品名を求める。
- 第3図のように、日付から新規入会までを印字する。なお、入会年がキャンペーン実施年と同じ場合は、新規入会に「○」を印字する。
- 粗品別の配付数をテーブル SSU-T に集計する。なお、SSU-T はテーブル KEIKA-T、テーブル SMEI-T と添字で対応している。

テーブル SSU-T

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| SSU | (1) | (2) | (3) | (4) | (5) |
|     |     |     |     |     |     |

- 粗品配付件数を集計する。
- ファイルを読み終えたあと、次の処理を行う。
- 粗品配付件数を印字する。
- 粗品別配付数を印字する。
- マグカップの割合(%)を次の計算式で求め、印字する。

$$\text{マグカップの割合(\%)} = \text{マグカップ配付数} \times 100 \div \text{粗品配付件数} \quad (\text{小数第1位未満四捨五入})$$

## &lt; COBOL プログラム &gt;

WORKING-STORAGE SECTION.

}

```

01 JNEN          PIC 9(04).
01 NNEN          PIC 9(04).
01 KNEN          PIC 9(02).
01 N             PIC 9(01).
01 HKEN          PIC 9(03).
01 WARI          PIC 9(03)V9(01).
01 MEISAI1.
  02             PIC X(10) VALUE SPACE.
  02 M-HI        PIC 9(04).
  02             PIC X(08) VALUE SPACE.
  02 M-BAN       PIC 9(08).
  02             PIC X(08) VALUE SPACE.
  02 M-SMEI      PIC X(10).
  02             PIC X(09) VALUE SPACE.
  02 M-SINKI     PIC X(02).

```

}

```

01 MEISAI3.
  02 M-R OCCURS 5 TIMES.
    03          PIC X(09).
    03 M-SSU    PIC ZZ9.

```

}

PROCEDURE DIVISION.

```

P1. OPEN INPUT HAIFU-F OUTPUT OUT-F
  INITIALIZE E-FLG SSU-T HKEN MEISAI3
  DISPLAY "キャンペーン実施年を入力してください"
  ACCEPT JNEN
  PERFORM UNTIL E-FLG = 1
    READ HAIFU-F
    AT END
      MOVE 1 TO E-FLG
    NOT AT END
      COMPUTE NNEN = H-BAN / 10000
      COMPUTE KNEN = JNEN - NNEN
      MOVE 1 TO N
      PERFORM UNTIL  >= KNEN
        COMPUTE N = N + 1
      END-PERFORM
      MOVE H-HI TO M-HI
      MOVE H-BAN TO M-BAN
      MOVE SMEI(N) TO M-SMEI
      IF  = JNEN
        THEN
          MOVE "○" TO M-SINKI
        ELSE
          MOVE " " TO M-SINKI
      END-IF
      WRITE OUT-R FROM MEISAI1 AFTER 1
      COMPUTE SSU(N) = SSU(N) + 1
      COMPUTE HKEN = 
    END-READ
  END-PERFORM
  MOVE HKEN TO M-HKEN
  WRITE OUT-R FROM MEISAI2 AFTER 1
  PERFORM VARYING N FROM 1 BY 1 UNTIL N > 5
    MOVE  TO M-SSU(N)
  END-PERFORM
  WRITE OUT-R FROM MEISAI3 AFTER 2
  COMPUTE WARI ROUNDED = 
  MOVE WARI TO M-WARI
  WRITE OUT-R FROM MEISAI4 AFTER 1
  CLOSE HAIFU-F OUT-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2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) |
|     | 回   |     |     |     |     |

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| 【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2級

審査基準

|     |   |   |   |   |   |             |
|-----|---|---|---|---|---|-------------|
| 【1】 | 1 | 2 | 3 | 4 | 5 | 各2点<br>計10点 |
|     | ケ | コ | イ | キ | ウ |             |

|     |   |   |   |   |   |             |
|-----|---|---|---|---|---|-------------|
| 【2】 | 1 | 2 | 3 | 4 | 5 | 各2点<br>計10点 |
|     | エ | カ | ク | オ | ア |             |

|     |   |   |   |   |   |             |
|-----|---|---|---|---|---|-------------|
| 【3】 | 1 | 2 | 3 | 4 | 5 | 各2点<br>計10点 |
|     | ア | イ | イ | ア | ウ |             |

|    |
|----|
| 小計 |
| 30 |

|     |     |     |     |     |     |             |
|-----|-----|-----|-----|-----|-----|-------------|
| 【4】 | (1) | (2) | (3) | (4) | (5) | 各3点<br>計15点 |
|     | 1回  | 6   | 16  | 20  | ウ   |             |

|     |     |     |     |     |     |             |
|-----|-----|-----|-----|-----|-----|-------------|
| 【5】 | (1) | (2) | (3) | (4) | (5) | 各3点<br>計15点 |
|     | カ   | ク   | オ   | ア   | ケ   |             |

|     |     |     |     |     |     |             |
|-----|-----|-----|-----|-----|-----|-------------|
| 【6】 | (1) | (2) | (3) | (4) | (5) | 各3点<br>計15点 |
|     | エ   | イ   | コ   | キ   | ウ   |             |

|    |
|----|
| 小計 |
| 45 |

..... [Java] ・ [マクロ言語] ・ [COBOL] .....

[Java] (注) =, 演算子の前後の空白は問わない。

|     |   |
|-----|---|
| 【7】 | (1) t a n j o B i   |
|     | (2) s e i z a B i [ i ]                                       |
|     | (3) n y u j o K e i / 7                                       |
|     | (4) a t o r a k u K e i [ a t o r a k u C o d e ] + n y u j o |
|     | (5) h < = 4   |

[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) J i ( S o e )                   |
|     | (2) S y a s y u = 1                 |
|     | (3) D a i D ( m )                   |
|     | (4) D a i S ( S o e ) + 1           |
|     | (5) R y o k i n k e i + R y o k i n |

[COBOL]

|     |                                   |
|-----|-----------------------------------|
| 【7】 | (1) K E I K A ( N )               |
|     | (2) N N E N                       |
|     | (3) H K E N + 1                   |
|     | (4) S S U ( N )                   |
|     | (5) S S U ( 4 ) * 1 0 0 / H K E N |

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