

問1 次のプログラムの説明及びプログラムを読んで、 に入れる適切な字句を、解答群の中から選べ。

＜プログラムの説明＞

＜プログラム＞

・ コマンドラインから与えられた数値を、数値の範囲区分ごとに個数をカウントし表示するプログラムである。

1. コマンドラインから一つ以上の整数を指定して実行する。また、コマンドラインから与えられた値が整数ではない場合は、考慮しないものとする。
2. 数値の範囲区分は、次の表のとおりとする。

表 数値の範囲区分

5 未満
5 以上, 10 未満
10 以上, 30 未満
30 以上, 100 未満
100 以上, 300 未満
300 以上

3. コマンドラインから指定された整数について、上記の表の範囲区分ごとに個数を数え、集計後に範囲区分と共に範囲区分ごとの個数を表示する。

《実行例》

```
>java Q5 100 3 430 234 2340 34 809 4 90 23
5 未満 2
5 以上, 10 未満 0
10 以上, 30 未満 1
30 以上, 100 未満 2
100 以上, 300 未満 2
300 以上 3
```

```
1 public class Q5 {
2     public static void main(String[] args) {
3         int segment;
4         int[] shuukei =  (28) ;
5         for (String s : args) {
6             int value = Integer.parseInt(  (29) );
7             if (value < 5) {
8                 segment = 0;
9             } else if (value < 10) {
10                segment = 1;
11            } else if (value < 30) {
12                segment = 2;
13            } else if (value < 100) {
14                segment = 3;
15            } else if (value < 300) {
16                segment = 4;
17            } else {
18                 (30) ;
19            }
20            shuukei[segment]++;
21        }
22
23        String segmentName[] = { "5 未満", "5 以上, 10 未満",
24                                "10 以上, 30 未満", "30 以上, 100 未満",
25                                "100 以上, 300 未満", "300 以上" };
26        for (int i = 0; i <  (31) ; i++) {
27            System.out.println( segmentName[i] + " " +  (32) );
28        }
29    }
30 }
```

(28) の解答群

- ア int[5]
- イ int[6]
- ウ new int[5]
- エ new int[6]

(29) の解答群

- ア s
- イ segment
- ウ args
- エ args[0]

(30) の解答群

- ア segment = 5
- イ segment = 6
- ウ segment = value
- エ shuukei[6]++

(31) の解答群

- ア 5
- イ segment
- ウ value
- エ shuukei.length

(32) の解答群

- ア i
- イ shuukei
- ウ shuukei[i]
- エ args[i]

(28)	(29)	(30)	(31)	(32)
エ	ア	ア	エ	ウ

問 2 次のプログラムの説明及びプログラムを読んで、 に入れる適切な字句を、解答群の中から選べ。

＜プログラムの説明＞

このプログラムは、複数の商品の売上データを基に ABC 分析を行い、計算した累積構成比に基づき、それぞれの商品が A, B, C のいずれのランクに分類されるかを表示するプログラムである。

- 全商品のデータは、その名前、売上金額があらかじめプログラム内の配列変数にそれぞれ設定されている。
- 各商品の表示及び ABC のランク付けは、以下の手順で行う。
  - 全商品のデータを売上金額の降順に並べる。なお、今回はあらかじめ売上金額の降順に並べられたデータを使用する。
  - 商品の売上総合計から各商品の構成比を計算する。
  - 売上金額の大きい商品から順に、各項目（分類、累積構成比、構成比、商品名）の出力を行う。累積構成比は、売上金額の大きい商品からその商品までの売上金額の累計を、売上総合計で割った値である。
  - 分類は、その商品の一つ前までの累積構成比に基づいて、次のように求める。
    - 75%未満：A
    - 75%以上、90%未満：B
    - 90%以上：C

- このプログラムでは Q6 クラス、ABCAnalysis クラス、Item クラスを定義する。各クラスの役割は、次のとおりである。

Q6 クラス                   ：実行用クラス  
ABCAnalysis クラス   ：全商品のデータの設定、ABC のランク付け、表示をするクラス  
Item クラス                ：商品のデータを保持するクラス

- ABCAnalysis クラス、Item クラスにおけるメンバ変数とメソッドは、次のとおりである。

ABCAnalysis	
char group[]	(ABC 分類の表示名)
Item[] items	(全商品のデータを保持する。)
double total	(全商品の売上合計)
setData メソッド	(配列変数から全商品のデータを items へ設定し、構成比を計算する。)
analysis メソッド	(累積構成比を計算しながら ABC 分析の区分を表示する。)

Item	
String name	(商品の名前)
int value	(商品の売上)
double percentage	(商品の構成比)
メソッドはなし	

《実行例》

>java Q6			
分類	累積構成比	構成比	商品名
A	56.1%	56.1%	コーヒー
A	84.5%	28.4%	お茶
B	94.8%	10.3%	ジュース
C	97.8%	3.0%	炭酸飲料
C	99.0%	1.2%	ミネラルウォーター
C	99.6%	0.6%	スポーツドリンク
C	100.0%	0.4%	栄養ドリンク

問2（つづき）

&lt;プログラム&gt;

```

1 class ABCAnalysis {
2     char group[] = { 'A', 'B', 'C' };
3     Item[] items;
4     double total;
5
6     void setData(String[] itemNames, int[] itemValues) {
7         items = new Item[itemValues.length];
8         for (int i = 0; i < itemValues.length; i++) {
9             total += (33);
10            Item shouhinn = new Item();
11            shouhinn.name = itemNames[i];
12            shouhinn.value = itemValues[i];
13            items[i] = shouhinn;
14        }
15        for (Item item : items) {
16            (34) = item.value / total;
17        }
18    }
19
20    void analysis() {
21        int cumulativeValue = 0;
22        double cumulativePercentage = 0.0;
23        int groupIdX = 0;
24
25        System.out.println("分類\t 累積構成比\t 構成比\t 商品名");
26        for (Item item : items) {
27            if (cumulativePercentage >= 0.9) {
28                groupIdX = (35);
29            } else if (cumulativePercentage >= 0.75) {
30                groupIdX = 1;
31            }
32            cumulativeValue (36);
33            cumulativePercentage = cumulativeValue / total;
34            System.out.println( (37) + "\t"
35                + (int) (cumulativePercentage * 1000) / 10.0 + "%\t\t"
36                + (int) (item.percentage * 1000) / 10.0 + "%\t"
37                + item.name + "\t" );
38        }
39    }
40 }

```

```

41
42 class Item {
43     String name;
44     int value;
45     double percentage;
46 }
47
48 public class Q6 {
49     public static void main(String args[]) {
50         String bunruiMei[] = {
51             "コーヒー", "お茶", "ジュース",
52             "炭酸飲料", "ミネラルウォーター", "スポーツドリンク",
53             "栄養ドリンク" };
54         int uriageData[] = { 561, 284, 103, 30, 12, 6, 4 };
55
56         ABCAnalysis abc = new ABCAnalysis();
57         abc.setData(bunruiMei, uriageData);
58         abc.analysis();
59     }
60 }

```

(33) の解答群

- ア items[i]
- イ items[i].value
- ウ itemValues[i]
- エ shouhinn.value

(34) の解答群

- ア item
- イ item.percentage
- ウ percentage
- エ percentage[i]

(35) の解答群

- ア -2
- イ -1
- ウ 0
- エ 2

(36) の解答群

- ア = item.value
- イ = item.percentage
- ウ += item.value
- エ += item.percentage

(37) の解答群

- ア item
- イ item.value
- ウ group
- エ group[groupIdX]

(33)	(34)	(35)	(36)	(37)
ウ	イ	エ	ウ	エ