

問2 次の文章の空欄 **オ** ~ **ケ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

Sさんは、先生(T)との会話を参考にして、変数 **year** に与えられた西暦に対し、その和暦を求めるプログラムを考えてみた(図3)。

ここでは例として、和暦に変換する西暦を1995年としている。配列 **Era_list** と配列 **Era_name** に図1、図2で考えた西暦と元号の対応を設定し、両方の配列に共通する添字を表す変数 **era_index** を用意し、初期値として0を用いた。

はじめに、西暦と対応する元号を求めるために、対応表(表1)を上から順に確認していく流れを、(05)~(07)行目のような繰り返し文とした。その中で、変換したい西暦と対応表内の西暦を比較し、対応表内の西暦の後に変換したい西暦がある場合に **era_index** を現在参照している添字に更新するようにした。

次に、変数 **japanese_year** に和暦に変換したときの年数が計算され、さらに、上で求めた元号と年数を結合して表示するようにした。

実行してみると「平成 **ア** 年」が表示されたので、正しく動作していることがわかる。

```

(01) year = 1995
(02) Era_list = [1868, 1912, 1926, 1989, 2019]
(03) Era_name = ["明治", "大正", "昭和", "平成", "令和"]
(04) era_index = 0
(05) iを オ ながら繰り返す:
(06) | もし カ ならば:
(07) | | era_index = i
(08) japanese_year = year - キ + 1
(09) 表示する (Era_name[era_index], japanese_year, "年")

```

図3 西暦を和暦に変換するプログラム

T: プログラム(図3)ができたようですね。ただし、このプログラムはあくまでも簡単に考えたものなので、正確に和暦変換を行うためにはもうひと工夫必要になります。具体的には、元年のときには2つの元号が混在しているので、月日までを考慮して比較する必要がありますね。

S: 確かにそのとおりですね! 対応表(表1)には改元月日が記載されていて、年の途中で元号が変わっていることがわかります。例えば、このプログラムでは変数 **year** の値を2019として実行すると、**ク** と表示されますが、実際は、2019年4月30日までは **ケ** になるのですね。

表1 西暦と和暦の対応(再掲)

西暦	和暦	改元月日
1868年	明治元年	9月8日
1912年	大正元年	7月30日
1926年	昭和元年	12月25日
1989年	平成元年	1月8日
2019年	令和元年	5月1日

オ の解答群

- ① 1から5まで1ずつ増やし
- ② 0から4まで1ずつ増やし
- ③ 5から1まで1ずつ減らし
- ④ 4から0まで1ずつ減らし

カ の解答群

- ① **year** >= **Era_list**[i]
- ② **year** >= i
- ③ **year** <= **Era_list**[i]
- ④ **year** <= i

キ の解答群

- ① i
- ② **Era_name**[**era_index**]
- ③ **era_index**
- ④ **Era_list**[**era_index**]

ク・**ケ** の解答群

- ① 平成元年
- ② 平成31年
- ③ 令和元年
- ④ 令和1年

次に、年数は元年の西暦を基に計算することができるので、西暦 1995 年であれば、平成元年の 1989 年を基に、

$$1995 - 1986 + 1 = 7 \text{ (年)}$$

と計算することができる。よって、西暦 1872 年は、明治元年を参照して、

$$\textcircled{3} \quad 1872 - 1868 + 1 = 5 \text{ (年)}$$

……Iの(答)

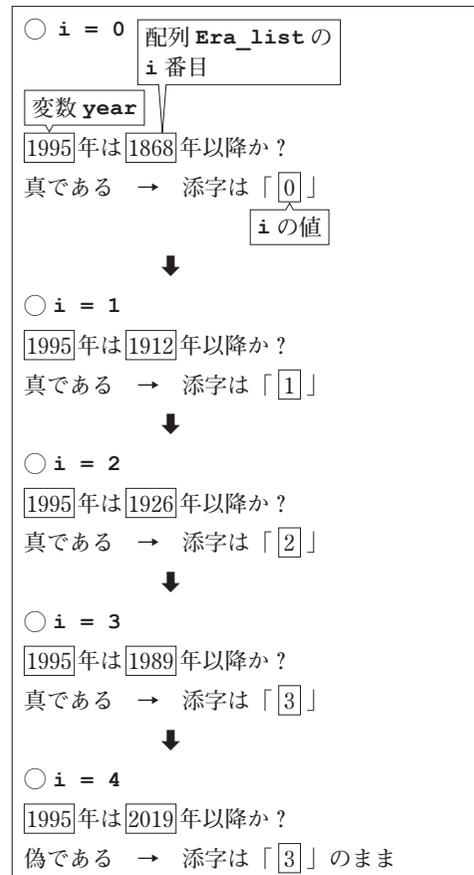
と計算すると明治 5 年であることがわかる。

問2 オ ② カ ① キ ③

ク ③ ケ ①

元年の西暦は `Era_list`、元号は `Era_name` に設定されている。変数 `year` に設定された西暦を和暦に変換するために、配列 `Era_list` の値を先頭から順番に変数 `year` と比較する。繰返しのための変数を `i` としたとき、配列 `Era_list` の要素数は 5 つであり、添字の先頭が 0 であることに注意すれば、**② 0 から 4 まで 1 つずつ増やし**ながら繰返せばよいとわかる。……オの(答)

繰返しの流れは以下のようになる。



繰返しの各回において、変数 `year` と `Era_list[i]` の値を比較しており、`year` が `Era_list[i]` の値以降であればよいので、以下のように表せる。

① `year >= Era_list[i]` ……カの(答)

これが真である場合、変数 `era_index` の値を `i` の値で更新していくので、最終的に `era_index` には変換に対応した位置の添字が入る。今回の場合、添字が「3」となったので西暦 1995 年の元号は `Era_name` の 3 番目を参照すれば `Era_name[3]` = "平成" であることがわかる。

年数の計算を式で表すと、

$$\text{変数 year } 1995 - \text{配列 Era_list の era_index 番目 } 1986 + 1 = 7$$

であるので、

$$\text{year} - \textcircled{3} \text{Era_list[era_index]} + 1 \quad \dots\dots \text{キの(答)}$$

となる。これを変数 `japanese_year` に設定する。これで元号を `Era_name[era_index]`、年数を `japanese_year` で表すことができたので、これらを並べて表示することで和暦変換を行うことができる。

ただし、このプログラムは「年単位」で変換を行っているため、月日まで考慮することができない。変数 `year` が 2019 である場合、繰返しは以下のようになる。



よって、変数 `era_index` の値は 4 であるため、元号は、

`Era_name[4]` = "令和"

年数は、

$$\text{year} - \text{Era_list}[4] + 1 = 2019 - 2019 + 1 = 1 \text{ (年)}$$

となる。これにより、表示上は「**③ 令和 1 年**」となるが、実際は改元月日が 5 月 1 日であるため、2019 年であっても「**① 平成 31 年**」である月日が存在している。……ク、ケの(答)

問3 コ ① サ ③ シ ②
ス ④ セ ① ソ ③
タ ⑥

関数「曜日確認 (西暦, 月, 日)」は、引数に設定された西暦, 月, 日の 3 つの値に応じて曜日を戻り値として返す関数となる。年月日から曜日を確認するための方法として、ツェラーの公式を利用しており、これは年月日を数式に入れて計算することで 0 から 6 の数値が得られ、これが日から土までの曜日と対応しているものである。プログラムで設定した配列 `Week_list` の添字と値の関係は以下のようになるので、`Week_list[3]` では **① 水** を取り出すことができる。……コの(答)

添字	0	1	2	3	4	5	6
値	日	月	火	水	木	金	土

このプログラムで注意しなければならないのは、1 月と 2 月の扱いである。ツェラーの公式に入れて計算する際には、前年の 13 月, 14 月とみなす必要があるため、以下の 2 つの処理が必要となる。

- ・西暦の値 (`year`) を 1 減じ前年の値にする
- ・月の値 (`month`) に 12 を加え 13 月, 14 月とする

これらの処理を行っているのはプログラムの以下の箇所である。

```

(03) もし サ ならば:
(04) | year = シ
(05) | month = ス
  
```

「1 月と 2 月のいずれか」という条件は、次のように表せる。

③ `month == 1 または month == 2`

……サの(答)

変数 `year` の値から 1 減じた値を新たに変数 `year` に設定するためには、次のようにすればよい。