

繰り返しfor文の復習問題①

1 次の仕様に基^{つぎ}づいて、プロ^{しょう}グラ^{もとづい}ムを^{ぶろぐらむ}作^{ざくせい}成^{さくせい}して^{さくせい}く^{さくせい}だ^{さくせい}さい。^{さくせい} (ファイル名^{ふあいるめい} : for1-01)

- ① 実行例^{じっこうれい}のとおり^{じっこうれい}に動^{どう}作^さする^さよう^さに、for文^{ふあーぶん}を使^{つか}い^{つか}プロ^{つか}グラ^{つか}ムを^{つか}作^{つか}っ^{つか}て^{つか}く^{つか}だ^{つか}さい。
(「***」を5回^{かいひょうじ}表^{かいひょうじ}示^{かいひょうじ}し^{かいひょうじ}ます。)

【実行例^{じっこうれい}】

```
>>> (executing file "for1-01.py")
***
***
***
***
***
```

2 次の仕様に基^{つぎ}づいて、プロ^{しょう}グラ^{もとづい}ムを^{ぶろぐらむ}作^{ざくせい}成^{さくせい}して^{さくせい}く^{さくせい}だ^{さくせい}さい。^{さくせい} (ファイル名^{ふあいるめい} : for1-02)

- ① 実行例^{じっこうれい}のとおり^{じっこうれい}に動^{どう}作^さする^さよう^さに、for文^{ふあーぶん}を使^{つか}い^{つか}プロ^{つか}グラ^{つか}ムを^{つか}作^{つか}っ^{つか}て^{つか}く^{つか}だ^{つか}さい。
た^{すう}だ^ちし、数^{ひょうじ}値^{ひょうじ}を^{ひょうじ}表^{ひょうじ}示^{ひょうじ}さ^{ひょうじ}せ^{ひょうじ}る^{ひょうじ}と^{ひょうじ}き^{ひょうじ}は、print関^{かん}数^{すう}の^{ひきすう}引^{ひきすう}数^{ひきすう}に^{へんすう}カ^{つか}ウ^{つか}ン^{つか}タ^{つか}変^{つか}数^{つか}を^{つか}使^{つか}う^{つか}よう^{つか}に^{つか}し^{つか}て^{つか}く^{つか}だ^{つか}さい。
(0から15ま^{すう}での^{ひょうじ}数^{ひょうじ}値^{ひょうじ}を^{ひょうじ}表^{ひょうじ}示^{ひょうじ}し^{ひょうじ}ます。)

【実行例^{じっこうれい}】

```
>>> (executing file "for1-02.py")
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

3 次の仕様に基^{つぎ}づいて、^{しよう}プログラムを^{もとづい}作成^{ふるくらむ}してください。^{さくせい} (ファイル名^{ふぁいるめい} : for1-03)

- ① 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふぁーぶん}するように、for文^{つか}を使い^{つく}プログラム^{つく}を作^{つく}ってください。
for1-02.pyのプログラムを、数値^{すうち}を表示^{ひょうじ}させるとき、print関数^{かんすう}の引数^{ひきすう}にカウンタ変数^{へんすう}を使^{つか}わないプログラム^かに変^かえてください。
(0から15までの数値^{すうち}を表示^{ひょうじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for1-03.py")
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

4 次の仕様に基^{つぎ}づいて、^{しよう}プログラムを^{もとづい}作成^{ふるくらむ}してください。^{さくせい} (ファイル名^{ふぁいるめい} : for1-04)

- ① 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふぁーぶん}するように、for文^{つか}を使い^{つく}プログラム^{つく}を作^{つく}ってください。
ただし、数値^{すうち}を表示^{ひょうじ}させるとき、print関数^{かんすう}の引数^{ひきすう}にカウンタ変数^{へんすう}を使^{つか}うようにしてください。
(-1から-10までの数値^{すうち}を表示^{ひょうじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for1-04.py")
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
```

5 次の仕様に基^{つぎ}づく^{しよう}いて、^{もとづい}プログラム^{ふるくらむ}を作成^{さくせい}してください。^ふ（^あファイル名^い：for1-05）^る

- ① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ふおーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
for1-04.pyのプログラムを、数値^{すうち}を表示^{ひょうじ}させるときに、print関数^{かんすう}の引数^{ひきすう}にカウンタ変数^{へんすう}を使^{つか}わないプログラム^かに変^かえてください。
（-1から-10までの数値^{すうち}を表示^{ひょうじ}します。）

【実行例^{じっこうれい}】

```
>>> (executing file "for1-05.py")
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
```

6 次の仕様に基^{つぎ}づく^{しよう}いて、^{もとづい}プログラム^{ふるくらむ}を作成^{さくせい}してください。^ふ（^あファイル名^い：for1-06）^る

- ① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ふおーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
（「★」を使^{ほし}って、実行例^{じっこうれい}のよう^{つか}な四角形^{しかくけい}を表示^{ひょうじ}します。）

【実行例^{じっこうれい}】

```
>>> (executing file "for1-06.py")
★★★★★
★★★★★
★★★★★
★★★★★
★★★★★
```

7 次の仕様に基^{つぎ}づいて、プロ^{しやう}グラ^{もとづい}ムを^{ふるく}作成^{らむ}して^{さくせい}ください。(ファイル名^ふ : for1-07)^{あいるめい} ^{ぶおー}

- ① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ぶあーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
(「■」^{しかく}を使って、実行例^{じっこうれい}のよう^{さんかっけい}な三角^{ひょうじ}形^じを^{ひょうじ}表示^じします。)

【実行例】

```
>>> (executing file "for1-07.py")
■
■ ■
■ ■ ■
■ ■ ■ ■
■ ■ ■ ■ ■
```

8 次の仕様に基^{つぎ}づいて、プロ^{しやう}グラ^{もとづい}ムを^{ふるく}作成^{らむ}して^{さくせい}ください。(ファイル名^ふ : for1-08)^{あいるめい} ^{ぶおー}

- ① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ぶあーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
(「▲」^{さんかく}を使って、実行例^{じっこうれい}のよう^{さんかっけい}な三角^{ひょうじ}形^じを^{ひょうじ}表示^じします。)

【実行例】

```
>>> (executing file "for1-08.py")
▲▲▲▲▲
▲▲▲▲
▲▲▲
▲▲
▲
```

9 次の仕様に基^{つぎ}づいて、プロ^{しやう}グラ^{もとづい}ムを^{ふるく}作成^{らむ}して^{さくせい}ください。(ファイル名^ふ : for1-09)^{あいるめい} ^{ぶおー}

- ① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ぶあーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
(数字^{すうじ}を使って、実行例^{じっこうれい}のよう^{ひょうじ}に表示^{すうじ}します。数字^{すうじ}は文字^{もじ}に変換^{へんかん}しましょう。)

【実行例】

```
>>> (executing file "for1-09.py")
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

- 10 次の仕様に基^{つぎ}づく^{しよう}いて、プ^{もとづい}ロ^{ふる}グ^{くら}ラ^むムを^{さくせい}作^ふ成^あして^{いる}く^{めい}だ^ささい。(フ^ふァ^あィ^いル^る名^{めい} : for1-10)
- ① 実^じ行^{こう}例^{れい}のとおり^じに動^{どう}作^さする^じよう^じに、for文^ふを^あい^ぶん^んをつ^{つか}か^{つか}い^{つか}つ^{つく}く^{つく}プログラム^{つく}を作^{つく}つ^{つく}て^{つく}く^{つく}だ^{つく}さい。
(数^{すう}字^じを^{すう}使^じつ^じて、実^じ行^{こう}例^{れい}の^じよう^じに表^ひ示^{ょう}し^じま^す。数^{すう}字^じは文^{もん}字^じに^{すう}変^も換^じし^まし^まう^す。

【実行例】

```
>>> (executing file "for1-10.py")
1
21
321
4321
54321
654321
7654321
87654321
987654321
```

- 11 次の仕様に基^{つぎ}づく^{しよう}いて、プ^{もとづい}ロ^{ふる}グ^{くら}ラ^むムを^{さくせい}作^ふ成^あして^{いる}く^{めい}だ^ささい。(フ^ふァ^あィ^いル^る名^{めい} : for1-11)
- ① 実^じ行^{こう}例^{れい}のとおり^じに動^{どう}作^さする^じよう^じに、for文^ふと^あを^あい^ぶん^んをつ^{つか}か^{つか}い^{つか}つ^{つく}く^{つく}プログラム^{つく}を作^{つく}つ^{つく}て^{つく}く^{つく}だ^{つく}さい。
た^{かん}だ^{すう}し、range関^{たい}数^{ひき}の^{すう}第^じ3引^{ひき}数^{すう}(ス^すテ^てッ^つプ)を^す使^じつ^じて^す作^{つく}り^{つく}ま^すし^まう^す。
(0^{すう}か^すら^す15^すま^すでの^す数^す値^じの^すう^すち、奇^き数^{すう}を^ひ表^ひ示^{ょう}し^ませ^ます。

【実行例】

```
>>> (executing file "for1-11.py")
1
3
5
7
9
11
13
15
```

- 12 次の仕様に基^{つぎ}づいて、プログラム^{しやう}を作成^{もとづい}してください。(ファイル名^{ふるくらむ} : for1-12)
① 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふあぶん}するように、for文^{つか}とを使い^{つか}プログラム^{つく}を作^{つく}ってください。
for1-11.pyのプログラム^{かんすう}について、range関数^{だい}は、第1引数^{ひきすう}だけ^{つか}を使って^{どうさ}動作^{どうさ}する
ように^か変^かえて^かください。
(0から15^{すうち}までの数値^{きすう}のうち、奇数^{ひょうじ}を表示^{ひょうじ}させます。)

【実行例】

```
>>> (executing file "for1-12.py")
1
3
5
7
9
11
13
15
```

- 13 次の仕様に基^{つぎ}づいて、プログラム^{しやう}を作成^{もとづい}してください。(ファイル名^{ふるくらむ} : for1-13)
① 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふあぶん}するように、for文^{つか}とを使い^{つか}プログラム^{つく}を作^{つく}ってください。
ただし、range関数^{かんすう}の第3引数^{だい}(ステップ^{ひきすう})^{つか}を使^{つか}わない^{つく}で作り^{つく}ましょう
(0から-20^{すうち}までの数値^{ぐうすう}のうち、偶数^{ひょうじ}だけ^{ひょうじ}を表示^{ひょうじ}させます。)

【実行例】

```
>>> (executing file "for1-13.py")
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
```

- 14 次つぎの仕様しように基づいてもとづい、プログラムふるくらむを作成さくせいしてください。(ファイル名ふぁいるめい : for1-14)
- ① 実行例じっこうれいのとおりどうさに動作ふぁーぶんするように、for文つかとを使いつかプログラムつくを作ってください。
for1-13.pyのプログラムについて、range関数かんすうは、第1引数だい ひきすうだけを使ってつか動作どうさするようにか変えてください。
(0から-20すうちまでの数値ぐうすうのうち、偶数ひょうじだけを表示ひょうじさせます。)

【実行例じっこうれい】

```
>>> (executing file "for1-14.py")
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
```

繰り返しfor文の復習問題②

1 次の仕様に基^つづいて、プログラムを作成^{さくせい}してください。(ファイル名: for2-01)

- ① input()関数^{かんすう}を利用して、キーボードから「名前」を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input name=>」とメッセージを表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふぁーぶん}するように、for文^{つか}を使いプログラム^{つく}を作^{つく}ってください。
(入力^{にゅうりよく}した名前^{なまえ}を8回^{かいひょうじ}表示^{ひょうじ}します。)

【実行例】

```
>>> (executing file "for2-01.py")
input name=>Azuma
Azuma
Azuma
Azuma
Azuma
Azuma
Azuma
Azuma
Azuma
```

2 次の仕様に基^つづいて、プログラムを作成^{さくせい}してください。(ファイル名: for2-02)

- ① input()関数^{かんすう}を利用して、キーボードから数値^{すうち}を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input number=>」とメッセージを表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふぁーぶん}するように、for文^{つか}を使いプログラム^{つく}を作^{つく}ってください。
(入力^{にゅうりよく}した数値^{すうち}の個数^{こすう}だけ、「●○●」を表示^{ひょうじ}します。)

【実行例】

```
>>> (executing file "for2-02.py")
input number=>5
●○●
●○●
●○●
●○●
●○●
```

```
>>> (executing file "for2-02.py")
input number=>7
●○●
●○●
●○●
●○●
●○●
●○●
●○●
```

3 次の仕様に基^{つぎ}づいて、プログラムを作成^{しよう}してください。(ファイル名^{もとづい} : for2-03)

- ① input^{いんぷつと}()関数^{かんすう}を利用して、キーボード^{りよう}から任意^{にんい}の単語^{たんご}を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input word=>」とメッセージ^{ひょうじ}を表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふあぶん}するように、for文^{つか}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
(入力^{にゅうりよく}した単語^{たんご}の文字数^{もじすう}だけ、その単語^{たんご}を表示^{ひょうじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for2-03.py")
input word=>apple
apple
apple
apple
apple
apple
```

```
>>> (executing file "for2-03.py")
input word=>strawberry
strawberry
```

4 次の仕様に基^{つぎ}づいて、プログラムを作成^{しよう}してください。(ファイル名^{もとづい} : for2-04)

- ① input^{いんぷつと}()関数^{かんすう}を利用して、キーボード^{りよう}から数値^{すうち}を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input number=>」とメッセージ^{ひょうじ}を表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふあぶん}するように、for文^{つか}を使い^{つか}プログラム^{つく}を作^{つく}ってください。
ただし、range関数^{かんすう}の第3引数^{だい ひきすう}(ステップ^{つか})を使^{つか}って作り^{つく}ましょう。
(入力^{にゅうりよく}した数値^{すうち}から30まで、三つ^{みつ}とばしで数^{かず}を表示^{ひょうじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for2-04.py")
input number=>5
5
8
11
14
17
20
23
26
29
```

```
>>> (executing file "for2-04.py")
input number=>9
9
12
15
18
21
24
27
30
```

つぎ しよう ちとづい ぶろくらむ さくせい ぶあいるめい ぶおー
5 次の仕様に基^つづいて、プログラム^を作成^して^くだ^さい。(ファイル名^は : for2-05)

- ① input^()関^数を利用^して、キーボード^から数^値を^入力^しま^す。
数^値を^入力^するときに、「input number=>」とメッ^セージ^を表^示し^ます。
- ② 実行例^のと^おりに動^作する^ように、for文^を使^いてプログラム^を作^って^くだ^さい。
た^だし、range関^数は第1引^数だけ^を使^って作^りま^しよ^う。
(入^力した数^値から、4を足^した数^値ま^での数の2乗^{した}数^を表^示し^ます。)

じっこうれい
【実行例】

```
>>> (executing file "for2-05.py")
input number=>1
1
4
9
16
25

>>> (executing file "for2-05.py")
input number=>5
25
36
49
64
81
```

```
>>> (executing file "for2-05.py")
input number=>-3
9
4
1
0
1

>>> (executing file "for2-05.py")
input number=>-9
81
64
49
36
25
```

6 次の仕様に基^{つぎ}づいて、プロ^{しよう}グラ^{もとづい}ムを^{ぶろくらむ}作成^{さくせい}して^ふくださ^あい。^い (ファイル名^ふ : for2-06)

- ① input()関^{いんぷつと}数^{かんすう}を^{りよう}利用^{すうち}して、キー^{にゅうりよく}ボ^{すうち}ード^{にゅうりよく}から^{すうち}数^{にゅうりよく}値^{にゅうりよく}を^{にゅうりよく}入^{にゅうりよく}力^{にゅうりよく}し^{にゅうりよく}ます。
数^{にゅうりよく}値^{にゅうりよく}を^{にゅうりよく}入^{にゅうりよく}力^{にゅうりよく}す^{にゅうりよく}と^{にゅうりよく}き^{にゅうりよく}に^{にゅうりよく}、「input number=>」と^{ひようじ}メ^{ひようじ}ッ^{ひようじ}セ^{ひようじ}ー^{ひようじ}ジ^{ひようじ}を^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ます。
- ② 実行^{じっこうれい}例^{じっこうれい}の^{じっこうれい}と^{じっこうれい}お^{じっこうれい}り^{じっこうれい}に^{じっこうれい}動^{ぶあーぶん}作^{つか}する^{つか}よ^{つか}う^{つか}に^{つか}、for文^{ふあーぶん}を^{つか}使^{つか}い^{つか}プロ^{つか}グラ^{つか}ムを^{つか}作^{つか}っ^{つか}て^{つか}くださ^{つか}い。
for2-05.pyで^{つか}作^{つか}った^{つか}プロ^{つか}グラ^{つか}ム^{つか}につ^{つか}いて^{つか}、range関^{かんすう}数^{だい}に^{だい}第^{だい}1引^{だい}数^{ひきすう}と^{ひきすう}第^{ひきすう}2引^{ひきすう}数^{つか}を^{つか}使^{つか}った^{つか}プロ^{つか}グラ^{つか}ム^{つか}に^{つか}変^{つか}え^{つか}て^{つか}くださ^{つか}い。
(入^{にゅうりよく}力^{すうち}した^{すうち}数^た値^{すうち}から^{かす}、4を^{じよう}足^{かす}した^{ひようじ}数^{ひようじ}値^{ひようじ}ま^{ひようじ}での^{ひようじ}数^{ひようじ}の^{ひようじ}2乗^{ひようじ}した^{ひようじ}数^{ひようじ}を^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ます。)

【実行例】

<pre>>>> (executing file "for2-06.py") input number=>1 1 4 9 16 25 >>> (executing file "for2-06.py") input number=>5 25 36 49 64 81</pre>	<pre>>>> (executing file "for2-06.py") input number=>-3 9 4 1 0 1 >>> (executing file "for2-06.py") input number=>-9 81 64 49 36 25</pre>
--	--

7 次の仕様に基^{つぎ}づいて、プロ^{しよう}グラ^{もとづい}ムを^{ぶろくらむ}作成^{さくせい}して^ふくださ^あい。^い (ファイル名^ふ : for2-07)

- ① input()関^{いんぷつと}数^{かんすう}を^{りよう}利用^{すうち}して、キー^{にゅうりよく}ボ^{すうち}ード^{にゅうりよく}から^{すうち}数^{にゅうりよく}値^{にゅうりよく}を^{にゅうりよく}入^{にゅうりよく}力^{にゅうりよく}し^{にゅうりよく}ます。
数^{にゅうりよく}値^{にゅうりよく}を^{にゅうりよく}入^{にゅうりよく}力^{にゅうりよく}す^{にゅうりよく}と^{にゅうりよく}き^{にゅうりよく}に^{にゅうりよく}、「input number=>」と^{ひようじ}メ^{ひようじ}ッ^{ひようじ}セ^{ひようじ}ー^{ひようじ}ジ^{ひようじ}を^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ます。
- ② 実行^{じっこうれい}例^{じっこうれい}の^{じっこうれい}と^{じっこうれい}お^{じっこうれい}り^{じっこうれい}に^{じっこうれい}動^{ぶあーぶん}作^{つか}する^{つか}よ^{つか}う^{つか}に^{つか}、for文^{ふあーぶん}を^{つか}使^{つか}い^{つか}プロ^{つか}グラ^{つか}ムを^{つか}作^{つか}っ^{つか}て^{つか}くださ^{つか}い。
(1から入^{にゅうりよく}力^{すうち}した^{すうち}数^{ごうけい}値^{ひようじ}ま^{ひようじ}での^{ひようじ}合^{ひようじ}計^{ひようじ}を^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ます。)

【実行例】

```
>>> (executing file "for2-07.py")
input number=>5
15

>>> (executing file "for2-07.py")
input number=>10
55

>>> (executing file "for2-07.py")
input number=>20
210
```

8 次の仕様に基^{つぎ}づいて、プロ^{しよう}グラ^{もとづい}ムを^{ふるくらむ}作成^{さくせい}して^ふください。^あ（^いファイル名^る：for2-08）

- ① input^{いんぷつと}()関^{かんすう}数^{りよう}を利用して、キー^{すう}ボ^ちード^{にゅうりよく}から数^{すう}値^ちを入^{にゅうりよく}力^ちします。
数^{すう}値^ちを入^{にゅうりよく}力^ちする^{ひようじ}と^じきに、「input number=>」とメ^{ひようじ}ッ^じセ^じー^じジ^じを^{ひようじ}表^じ示^じし^じま^じす。
- ② 実行^{じっこうれい}例^{れい}の^{じっこうれい}と^{じっこうれい}お^{じっこうれい}り^{じっこうれい}に^{じっこうれい}動^{どう}作^さする^{じっこうれい}よ^{じっこうれい}う^{じっこうれい}に^{じっこうれい}、for^{ふあーぶん}文^{ぶん}を^{じっこうれい}使^{つか}い^{じっこうれい}プロ^{じっこうれい}グラ^{じっこうれい}ム^{じっこうれい}を^{じっこうれい}作^{じっこうれい}っ^{じっこうれい}て^{じっこうれい}く^{じっこうれい}だ^{じっこうれい}さ^{じっこうれい}い。
（入^{にゅうりよく}力^ちした^{すう}数^ちの^{すう}文^も字^じ数^{すう}だ^もけ^じ、^も文^も字^じ列^{れつ}「123451234512345…」を^{ひようじ}表^じ示^じし^じま^じす。）

【^{じっこうれい}実行^{れい}例^{れい}】

```
>>> (executing file "for2-08.py")
input number=>5
12345

>>> (executing file "for2-08.py")
input number=>8
12345123

>>> (executing file "for2-08.py")
input number=>9
123451234

>>> (executing file "for2-08.py")
input number=>13
1234512345123
```

9 次の仕様に基^{つぎ}づいて、プロ^{しよう}グラ^{もとづい}ムを^{ふるくらむ}作成^{さくせい}してください。(ファイル名^{ふあいるめい} : for2-09)

- ① input^{いんぷット}()関^{かんすう}数^{りよう}を利用して、キー^{すうち}ボ^{にゆうりよく}ードから「列^{れつすう}数^{ぎようすう}」と「行^{にゆうりよく}数^{ぎようすう}」を入^{にゆうりよく}力^{にゆうりよく}します。
数^{すうち}値^{にゆうりよく}を入^{にゆうりよく}力^{にゆうりよく}するときに、「column number=>」「row number=>」とメ^{ひようじ}ッセージ^{ひようじ}を^{ひようじ}表^{ひようじ}示^{ひようじ}します。
- ② 実^{じっこうれい}行^{じっこうれい}例^{じっこうれい}のとおり^{じっこうれい}に動^{どうさ}作^{どうさ}する^{どうさ}よう^{どうさ}に、for^{ふおーぶん}文^{つか}を使^{つか}い^{つか}プロ^{つか}グラ^{つか}ムを^{つか}作^{つか}つて^{つか}くだ^{つか}さい。
(入^{にゆうりよく}力^{にゆうりよく}した^{にゆうりよく}行^{にゆうりよく}数^{にゆうりよく}分^{にゆうりよく}、入^{にゆうりよく}力^{にゆうりよく}した^{にゆうりよく}列^{れつすうぶん}数^{しかく}分^{ひようじ}の「◇」と表^{ひようじ}示^{ひようじ}します。)

【実行例】

```
>>> (executing file "for2-09.py")
column number=>5
row number=>8
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇

>>> (executing file "for2-09.py")
column number=>10
row number=>5
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
```

く かえ ふおーぶん ふくしゅうもんだい
繰り返しfor文の復習問題③

つぎ しよう もとづい ぶろぐらむ さくせい ふあいるめい ふおー
1 次の仕様に基づいて、プログラムを作成してください。(ファイル名: for3-01)

じっこうれい どうさ ふおーぶん つか つく
① 実行例のとおり動作するように、for文を使いプログラムを作ってください。

ただし、if文を使ってプログラムを作りましょう。

にゅうりよく すうち すうち ひょうじ
(入力した数値から-20までの数値のうち、偶数だけを表示させます。)

じっこうれい
【実行例】

```
>>> (executing file "for3-01.py")
input number=>2
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
```

```
>>> (executing file "for3-01.py")
input number=>5
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
```

```
>>> (executing file "for3-01.py")
input number=>-10
-10
-12
-14
-16
-18
-20
```


4 以下の仕様に基づいて、プログラムを作成してください。(ファイル名: for3-04)

- ① input()関数を利用して、「正の整数」を入力します。
数値を入力するときに、「input positive number=>」とメッセージを表示します。
- ② 実行例のとおり動作するように、for文を使いプログラムを作ってください。
(0から入力した数値まで表示させます。このとき、偶数のときには「even」、奇数のときには「odd」と表示します。0より小さい数字を入力した場合は、「invalid!」と表示します。)

【実行例】

```
>>> (executing file "for3-04.py")
input positive number=>10
0 even
1 odd
2 even
3 odd
4 even
5 odd
6 even
7 odd
8 even
9 odd
10 even

>>> (executing file "for3-04.py")
input positive number=>-5
invalid!
```

5 次の仕様に基^{つぎ}づいて、プログラムを作成^{しやう}してください。(ファイル名^{もとづい} : for3-05)

① input()関^{いんぷと}数^{かんすう}を利用して、^{りよう}「負^ふの整数^{せいすう}」を^{にゅうりよく}入^{にゅうりよく}力^{りよく}します。
数^{すう}値^ちを^{にゅうりよく}入^{にゅうりよく}力^{りよく}する^{にゅうりよく}と^{にゅうりよく}き^{にゅうりよく}に、^{にゅうりよく}「input negative number=>」と^{にゅうりよく}メ^{にゅうりよく}ッ^{にゅうりよく}セ^{にゅうりよく}ー^{にゅうりよく}ジ^{にゅうりよく}を^{にゅうりよく}表^{にゅうりよく}示^{にゅうりよく}し^{にゅうりよく}ま^{にゅうりよく}す。

② 実行例^{じっこうれい}の^{じっこうれい}と^{じっこうれい}お^{じっこうれい}り^{じっこうれい}に^{じっこうれい}動^{どう}作^さする^{じっこうれい}よ^{じっこうれい}う^{じっこうれい}に、^{じっこうれい}for^{じっこうれい}文^{ぶん}を^{じっこうれい}使^{つか}い^{じっこうれい}プ^{じっこうれい}ロ^{じっこうれい}グ^{じっこうれい}ラ^{じっこうれい}ム^{じっこうれい}を^{じっこうれい}作^{つく}っ^{じっこうれい}て^{じっこうれい}く^{じっこうれい}だ^{じっこうれい}さ^{じっこうれい}い。

(-1から入^{にゅうりよく}力^{りよく}した^{にゅうりよく}数^{すう}値^ちま^{ひょうじ}で^{ひょうじ}表^{ひょうじ}示^しさ^しせ^しま^しす。
この^{ばいすう}と^しき、^し2^しの^し倍^{ばい}数^{すう}の^しと^しな^しり^しに^しは^し「◆」、^し3^しの^し倍^{ばい}数^{すう}の^しと^しな^しり^しに^しは^し「◇」、^し6^しの^し倍^{ばい}数^{すう}の^しと^しな^しり^しに^しは^し「◆◇」、^しと^し表^し示^しま^しす。^し0^し以^し上^しの^し数^{すう}字^じを^し入^{にゅうりよく}力^{りよく}した^し場^ば合^{あい}は、
「invalid!」と^{ひょうじ}表^{ひょうじ}示^しま^しす。)

【実行例】

```
>>> (executing file "for3-05.py")
input negative number=>-15
-1
-2 ◆
-3 ◇
-4 ◆
-5
-6 ◆◇
-7
-8 ◆
-9 ◇
-10 ◆
-11
-12 ◆◇
-13
-14 ◆
-15 ◇

>>> (executing file "for3-05.py")
input negative number=>5
invalid!
```

6 次の仕様に基^{つぎ}づいて、プログラムを作成^{しやう}してください。(ファイル名^{もとづい} : for3-06)

- ① input^{いんぷット}()関数^{かんすう}を利用して、キーボードから「列数^{りやう}」と「行数^{れつすう}」を入力^{ぎやうすう}します。数値^{すうち}を入力^{にゆうりよく}するときに、「column number=>」「row number=>」とメッセージ^{ひやうじ}を表示^{ひやうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{どうさ}するように、for文^{ふおーぶん}を使い^{つか}プログラム^{つく}を作^{つく}ってください。(「■」と「□」を使^{つか}って、入^い力^{りき}した列数^{れつすう}、行^{ぎやうすう}数^{すう}分^{ぶん}の四角形^{しかくけい}を^{ひやうじ}表示^{ひやうじ}します。頂^{ちやうてん}点^{てん}には「■」を^{しかく}表示^{ひやうじ}させるように^いします。い^いずれも0^{すうち}以下^{にゆうりよく}の数値^{ばあい}を^{ひやうじ}入^い力^{りき}した場^{ばあい}合^あは、「invalid!」と^{ひやうじ}表示^{ひやうじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for3-06.py")
column number=>3
row number=>3
■ □ ■
□ □ □
■ □ ■

>>> (executing file "for3-06.py")
column number=>5
row number=>4
■ □ □ □ ■
□ □ □ □ □
□ □ □ □ □
■ □ □ □ ■
```

```
>>> (executing file "for3-06.py")
column number=>1
row number=>0
invalid!

>>> (executing file "for3-06.py")
column number=>-3
row number=>4
invalid!

>>> (executing file "for3-06.py")
column number=>0
row number=>0
invalid!
```

7 次の仕様に基^{つぎ}づいて、プログラム^{しよう}を作成^{もとづい}してください。(ファイル名^{ふるくらむ} : for3-07)

- ① input^{いんぷット}()関数^{かんすう}を利用して^{りよう}、キーボード^{ぎようすう}から「行数^{にゆうりよく}」を入力^{すうち}します。
数値^{すうち}を入力^{にゆうりよく}するときに、「row number =>」とメッセージ^{ひようじ}を表示^{ひようじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ぶあーぶん}するように、for文^{つか}を使い^{つく}プログラム^{つく}を作^{つく}ってください。
(入力^{にゆうりよく}した行数^{ぎようすうぶん}分^{さんかく}、「▲」^{さんかく}と「△」^{さんかく}で作^{つく}った三角形^{さんかくけい}を表示^{ひようじ}します。ただしこのとき、1行目^{ひようじ}は「▲」^{さんかく}を1つ、2行目^{ひようじ}は「△」^{さんかく}を2つ、3行目^{ひようじ}は「▲」^{さんかく}を3つ…というように表示^{ひようじ}します。また、行数^{ぎようすう}に0以下^いの数値^{すうち}を入力^{にゆうりよく}した場合は^{あい}、「invalid!」^{ひようじ}と表示^{ひようじ}します。)

【実行例^{じっこうれい}】

```
>>> (executing file "for3-07.py")
row number=>10
▲
△△
▲▲▲
△△△△
▲▲▲▲▲
△△△△△△
▲▲▲▲▲▲▲
△△△△△△△△
▲▲▲▲▲▲▲▲▲
△△△△△△△△△△
```

```
>>> (executing file "for3-07.py")
row number=>4
▲
△△
▲▲▲
△△△△

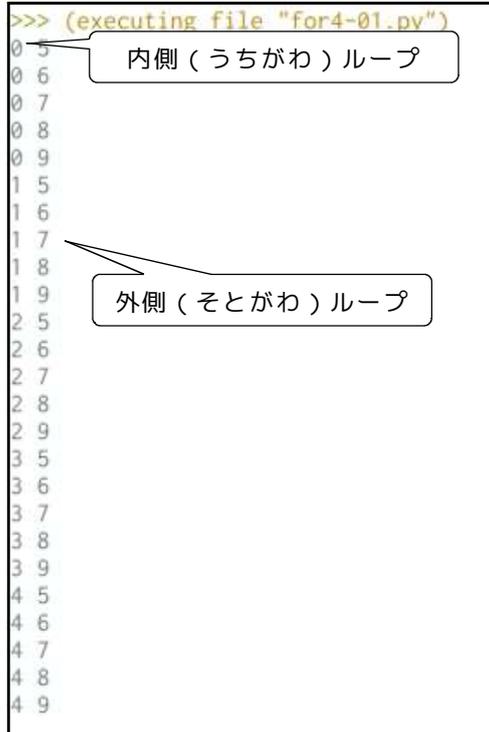
>>> (executing file "for3-07.py")
row number=>-1
invalid!
```

繰り返しfor文の復習問題④

- 1 次の仕様に基^{つぎ}づいて、プロ^ふグ^ろラ^むム^{さくせい}を作成^ふして^あく^いだ^さい^い。(ファイル名^ふ: for4-01)
① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^ふの二重ループ^おを使った^{つか}プログラム^{つく}を作^{つく}ってください。

【実行例】

```
>>> (executing file "for4-01.py")
0 5
0 6
0 7
0 8
0 9
1 5
1 6
1 7
1 8
1 9
2 5
2 6
2 7
2 8
2 9
3 5
3 6
3 7
3 8
3 9
4 5
4 6
4 7
4 8
4 9
```



2 次の仕様に基^{つぎ}づく^{しよう}いて、^{もとづい}プログラム^{ふるくらむ}を作成^{さくせい}してください。^ふ（^あファイル名^い：for4-02）^{ぶおー}

① 実行例^{じっこうれい}のとおり^{どう}に動作^さするように、for文^{ぶあーぶん}を使い^{つか}プログラム^{つか}を作^{つく}ってください。

（for4-01.pyを、for文^{ぶん}の2重^{じゅう}ループ^{つか}を使った^{つか}プログラム^{つか}を使^{つか}って書^かきかえます。

^{しゅつりやく}出力^{すうち}される^で数値^{れつ}のうち、3または5^{ひょうじ}が出る^で列^じは「***」と表示^{ひょうじ}します）

【実行例】

```
>>> (executing file "for4-02.py")
***
0 6
0 7
0 8
0 9
***
1 6
1 7
1 8
1 9
***
2 6
2 7
2 8
2 9
***
***
***
***
***
4 6
4 7
4 8
4 9
```

3 次の仕様に基^{つぎ}づいて、プログラムを作成^{もとづい}してください。(ファイル名^{ふろくらむ} : for4-03)

- ① input^{いんぷット}()関数^{かんすう}を利用して^{りよう}、キーボードから「文字数^{もじすう}」と「行数^{ぎょうすう}」を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「column number=>」「row number=>」とメッセージ^{ひょうじ}を表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふぉーぶん}するように、for文^{つか}を使い^{つか}プログラム^{つか}を作^{つく}ってください。
(for2-09で作成^{ふぉーぶん}したプログラム^{ぶん}を、for文^{ぶん}の2重^{じゅう}ループ^{つか}を使ったプログラム^{つか}に書き直^かしてください。また、どちらかに0が入力^{なほ}されたときに、「invalid!」と表示^{ひょうじ}されるようにしてください。)

【実行例^{じっこうれい}】

```
>>> (executing file "for4-03.py")
column number=>5
row number=>8
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇
◇◇◇◇◇

>>> (executing file "for4-03.py")
column number=>10
row number=>5
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
◇◇◇◇◇◇◇◇◇◇
```

```
>>> (executing file "for4-03.py")
column number=>-3
row number=>5
invalid!

>>> (executing file "for4-03.py")
column number=>0
row number=>2
invalid!

>>> (executing file "for4-03.py")
column number=>-9
row number=>-8
invalid!
```

4 次の仕様に基^{つぎ}づいて、プログラムを作成^{しよ}してください。(ファイル名: for4-04)

- ① input()関^{かん}数^{すう}を利用^{りよう}して、キーボードから数値^{すうち}を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input number=>」とメッセージを表示^{ひょうじ}します。
- ② 実行例^{じっこうれい}のとおり^{どう}に動作^{どうさ}するように、for文^{ふおーぶん}を使い^{つか}プログラムを作^{つく}ってください。
(for文^{ぶん}の2重^{じゅう}ループ^{るूप}を使^{つか}ったプログラムで作^{つく}ってください。入^{にゅうりよく}力^{すうち}した数値^{すうち}を使^{つか}い、三^{さん}角^{かく}形^{けい}をつくりま^{さん}す。た^だし、1から9以外^{いがい}の数値^{すうち}が入^{にゅうりよく}力^{すうち}されたときは、「input only 1 - 9」^{ひょうじ}と表示^{ひょうじ}します。)

【実行例】

```
>>> (executing file "for4-04.py")
input number=>4
4
43
432
4321

>>> (executing file "for4-04.py")
input number=>9
9
98
987
9876
98765
987654
9876543
98765432
987654321
```

```
>>> (executing file "for4-04.py")
input number=>0
input only 1 - 9

>>> (executing file "for4-04.py")
input number=>10
input only 1 - 9
```

- 5 次の仕様に基づいて、プログラムを作成してください。(ファイル名: for4-05)
- ① input()関数を利用して、キーボードから「列数」と「行数」を入力します。
数値を入力するときに、「column number=>」「row number=>」とメッセージを表示します。
 - ② 実行例のとおり動作するように、for文を使いプログラムを作ってください。
(for3-06で作成したプログラムを呼び出して、for文の2重ループを使ったプログラムに書き直してください)

【実行例】

<pre>>>> (executing file "for4-05.py") column number=>3 row number=>3 ■ □ ■ □ □ □ ■ □ ■ >>> (executing file "for4-05.py") column number=>5 row number=>4 ■ □ □ □ ■ □ □ □ □ □ □ □ □ □ □ ■ □ □ □ ■</pre>	<pre>>>> (executing file "for4-05.py") column number=>1 row number=>0 invalid! >>> (executing file "for4-05.py") column number=>-3 row number=>4 invalid! >>> (executing file "for4-05.py") column number=>0 row number=>0 invalid!</pre>
---	--

6 次の仕様に基^{つぎ}づいて、プログラムを作成^{しよ}してください。(ファイル名^も : for4-06)

- ① input()関^{いんぷつと}数^{かんすう}を利用して、キーボードから数^{りよう}値^{すう}を^ち入^{にゅうりよく}力^{りよく}します。
数^{すう}値^ちを^{にゅうりよく}入^{りよく}力^{りよく}するときに、「input number=>」とメッセ^{ひようじ}ージ^じを^{ひようじ}表^じ示^じします。
- ② 実行例^{じっこうれい}のとおり^{じっこうれい}に動^{どう}作^さする^{つか}よう^{つか}に、for文^{ふおーぶん}を使^{つか}い^{つか}プログラム^{つか}を^{つか}作^{つか}って^{つか}くださ^{つか}い。
た^{つか}だ^{つか}し、break^{つか}を使^{つか}って^{つか}作^{つか}って^{つか}くださ^{つか}い。
(for文^{つか}の2重^{ふん}ループ^{じゅう}を使^{つか}ったプログラム^{つか}で^{つか}作^{つか}って^{つか}くださ^{つか}い。入^{にゅうりよく}力^{すう}した^ち数^{すう}値^ちの^{すう}文字^{もじ}数^{すう}だ^{もじ}け、文字^{もじ}列^{もじ}「1234512345…」を^{ひようじ}表^じ示^じします。た^じだ^じし、1行^{ぎよう}5文字^{もじ}ま^{もじ}でと^{もじ}し、5文字^{もじ}ご^{もじ}とに改^{かい}行^{ぎよう}する^{もじ}ものとし^{もじ}ます。)

【実行例^{じっこうれい}】

```
>>> (executing file "for27.py")
input number=>5
12345

>>> (executing file "for27.py")
input number=>7
12345
12
1

>>> (executing file "for27.py")
input number=>16
12345
12345
12345
1
```

```
>>> (executing file "for27.py")
input number=>29
12345
12345
12345
12345
1234
```

7 次の仕様に基^{つぎ}づく^{しよう}いて、プログラム^{もとづい}を作成^{ふるくらむ}してください。(ファイル名^{さくせい} : for4-07)

① input()関数^{ふあいるめい}を利用して、キーボード^{ふあー}から数値^{すうち}を入力^{にゅうりよく}します。
数値^{すうち}を入力^{にゅうりよく}するときに、「input number=>」とメッセージ^{ひょうじ}を表示^{ひょうじ}します。

② 実行例^{じっこうれい}のとおり^{どうさ}に動作^{ふあーぶん}するように、for文^{つか}を使い^{つか}プログラム^{つか}を作^{つく}ってください。
ただし、break^{つか}を使^{つか}って作^{つく}ってください。

(for文^{ふん}の2重^{じゅう}ループ^{つか}を使^{つか}ったプログラム^{ひょうじ}で作^{つく}ってください。入^{にゅうりよく}力^{すうち}した数値^{すうち}の文字^{もじ}数^{すう}だけ、文字^{もじ}列^{れつ}「○●○○●○○…」^{まるまるまるまるまる}」^{ひょうじ}を表示^{ひょうじ}します。ただし、1行^{ぎょう}5文字^{もじ}までとし、5文字^{もじ}ごとに改行^{かいぎょう}するものとします。)

※プログラム^{かんせい}が完成^{あんもく}したら、「暗黙^{あんもく}のTrue, False」^{つか}を使^{つか}った文^{ぶん}に書^かきかえましょ
う

【実行例】

```
>>> (executing file "for4-07.py")
input number=>4
○●○○●

>>> (executing file "for4-07.py")
input number=>11
○●○○●○○
●○○●○○●
○
```

```
>>> (executing file "for4-07.py")
input number=>13
○●○○●○○
●○○●○○●
○●○○

>>> (executing file "for4-07.py")
input number=>20
○●○○●○○
●○○●○○●
○●○○●○○
○●○○●○○
```