

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^{もとづい}ラム^{ぶろぐらむ}を^{ざくせい}作成^ふして^あく^いださい。^{めい} (ファイル名^{ふあ} : for1-01)

- ① 実行例^{じっこうれい}のとおり^{どう}に動^さ作^{ふあ}する^{ぶん}よう^{つか}に、for文^{つか}を使^{つく}い^{つく}プロ^{つく}グ^{つく}ラム^{つく}を^{つく}作^{つく}って^{つく}く^{つく}ださい。
(「***」を5回表示^{かいひょうじ}します。)

【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^{もとづい}ラム^{ぶろぐらむ}を^{ざくせい}作成^ふして^あく^いださい。^{めい} (ファイル名^{ふあ} : for1-02)

- ① 実行例^{じっこうれい}のとおり^{どう}に動^さ作^{ふあ}する^{ぶん}よう^{つか}に、for文^{つか}を使^{つく}い^{つく}プロ^{つく}グ^{つく}ラム^{つく}を^{つく}作^{つく}って^{つく}く^{つく}ださい。
た^{すう}だ^ちし、数^{ひょうじ}値^{ひょうじ}を^{かん}表^{すう}示^{ひき}さ^{すう}せ^{すう}るとき^{へん}は、print関^{つか}数^{つか}の引^{へん}数^{すう}にカ^{つか}ウン^{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^{かん}表^{すう}示^{ひき}さ^{すう}します。)

【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。(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。(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。(ファイル名^{ふぁいるめい} : for1-05)

- 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^{すうち}を作^{ひょうじ}ってください。
for1-04.pyのプログラムを、数値^{かんすう}を表示^{ひきすう}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。(ファイル名^{ふぁいるめい} : for1-06)

- 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^{すうち}を作^{ひょうじ}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 次の仕様に基づいて、プログラムを作成してください。(ファイル名: for1-10)
- 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。
(数字を使って、実行例のように表示します。数字は文字に変換しましょう。)

【実行例】

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

- 11 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: for1-11)
- ① 実行例のとおり動作するように、for文とを使いプログラムを作ってください。
ただし、range関数の第3引数(ステップ)を使って作りましょう。
(0から15までの数値のうち、奇数を表示させます。)

【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^{しょう}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^あい。 (ファイル名^い : for1-12)
- ① 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^{つか}ように、for文^{つか}とを^{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。
for1-11.pyのプログラム^{かんすう}につ^{だい}いて、range関^{ひきすう}数は、第1引^{つか}数^{どうさ}だけ^{つか}を使^{どうさ}って^{どうさ}動作^{どうさ}する^{どうさ}ように^か変^かえて^かくださ^かい。
(0から15^{すうち}までの^{きすう}数^{ひょうじ}値^{ひょうじ}の^{ひょうじ}う^{ひょうじ}ち^{ひょうじ}、奇^き数^{すう}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}ま^{ひょうじ}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^{しょう}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^あい。 (ファイル名^い : for1-13)
- ① 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^{つか}ように、for文^{つか}とを^{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。
た^{かんすう}だ^{だい}し^{ひきすう}、range関^{つか}数^{つか}の^{つか}第^{つか}3^{つか}引^{つか}数^{つか} (ス^{つか}テ^{つか}ッ^{つか}プ^{つか})^{つか}を^{つか}使^{つか}わ^{つか}な^{つか}い^{つか}で^{つか}作^{つか}り^{つか}ま^{つか}し^{つか}よ^{つか}う
(0から-20^{すうち}までの^{きすう}数^{ひょうじ}値^{ひょうじ}の^{ひょうじ}う^{ひょうじ}ち^{ひょうじ}、偶^ふ数^{すう}だ^{ひょうじ}け^{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}ま^{ひょうじ}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動作つか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^{ふあーぶん}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^{ふあーぶん}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()関数^{いんぷつと かんすう}を利用して、キーボードから任意^{りよう}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
strawberry
strawberry
strawberry
strawberry
strawberry
strawberry
strawberry
strawberry
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()関数^{いんぷつと 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^{ふあぶん}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。
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}を使って作りま^{つく}しょう。
(入力^{にゅうりよく}した数値^{すうち}から30まで、三つとばし^{みつ かず}で数^{ひょうじ}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^{さくせい}してください。(ファイル名: for2-05)

- ① input()関^{かん}数^{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^{どお}に動作^{どうさ}するように、for文^{ぶあーぶん}を使い^{つか}プログラム^つを作^{つく}ってください。
ただし、range関^{かん}数^{すう}は第1引数^{だいひきすう}だけ^{つか}を使って^{つく}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、4を足^たした数値^{すうち}までの数^{かず}の2乗^{じよう}した数^{かず}を表示^{ひょうじ}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^{しやう}グラ^{もとづい}ムを^{ぶろく}作成^{らむ}して^{さくせい}ください。(ファイル名^ふ : for2-06)

- ① input()関^{いんぷつと}数^{かんすう}を利用^{りよう}して、キー^{すう}ボ^ちード^にか^らら数^に値^{ゆうりよく}を^を入^を力^をし^ます。
数^{すう}値^ちを^に入^を力^をす^とき^に、「input number=>」とメ^{ひよう}ッ^じセ^をー^をジ^をを^を表^を示^をし^ます。
- ② 実行^{じっこう}例^{れい}の^のと^おり^にに^どう^さ作^ぶ動^あ作^ぶす^るよ^うに^に、for文^{ふあーぶん}を^を使^{つか}い^をプロ^をグラ^をム^をを^を作^をっ^てて^くだ^さい^ない。
for2-05.pyで^で作^をっ^たた^たプロ^をグラ^をム^をにつ^いて^に、range関^{かん}数^{すう}に^に第^だ1^い引^ひ数^{きすう}と^と第^だ2^い引^ひ数^{きすう}を^を使^{つか}っ^たた^たプロ^をグラ^をム^をに^に変^をえ^てて^くだ^さい^ない。
(入^に力^{ゆうりよく}し^た数^{すう}値^ちか^ら、4を^を足^たし^た数^{すう}値^ちま^での^の数^{かず}の^の2乗^{じよう}し^た数^{かず}を^を表^{ひよう}示^じし^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^{しやう}グラ^{もとづい}ムを^{ぶろく}作成^{らむ}して^{さくせい}ください。(ファイル名^ふ : for2-07)

- ① input()関^{いんぷつと}数^{かんすう}を利用^{りよう}して、キー^{すう}ボ^ちード^にか^らら数^に値^{ゆうりよく}を^を入^を力^をし^ます。
数^{すう}値^ちを^に入^を力^をす^とき^に、「input number=>」とメ^{ひよう}ッ^じセ^をー^をジ^をを^を表^を示^をし^ます。
- ② 実行^{じっこう}例^{れい}の^のと^おり^にに^どう^さ作^ぶ動^あ作^ぶす^るよ^うに^に、for文^{ふあーぶん}を^を使^{つか}い^をプロ^をグラ^をム^をを^を作^をっ^てて^くだ^さい^ない。
(1から入^に力^{ゆうりよく}し^た数^{すう}値^ちま^での^の合^{ごう}計^{けい}を^を表^{ひよう}示^じし^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^る名^{めい}：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^ふください。^あ (フ^いァ^いル^る名^{めい} : for2-09)

- 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^にから「列^{れつ}数^{すう}」と「行^{ぎよう}数^{すう}」を入^に力^{ゆうりよく}します。
数^{すう}値^ちを入^に力^{ゆうりよく}する^すと^きに、「column number=>」「row number=>」とメ^ひッ^{よう}セ^じー^じジ^じを表示^{ひょうじ}します。
- ② 実行^{じっこう}例^{れい}の^どう^さに^ふお^ーぶ^んん^{つか}に^つか^かく^つく^くように、for文^{ふおーぶん}を使^{つか}い^{つか}プロ^つグラ^くム^つを^つく^くて^くだ^さい。
(入^に力^{ゆうりよく}した^ぎょう^{すう}ぶ^んん^にゆう^りよく^{れつ}すう^ぶん^{しかく}ひよう^じ、入^に力^{ゆうりよく}した^ぎょう^{すう}ぶ^んん^にゆう^りよく^{れつ}すう^ぶん^{しかく}ひよう^じの「◇」と表示^{ひょうじ}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 しょう 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。

にゅうりょく 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
```

2 次の仕様に基^{つぎ}づいて、プログラム^{しよ}を作成^{もとづい}してください。(ファイル名^{ぶろくらむ} : for3-02)

- ① input()関数^{いんぷつと}を利用して、キーボード^{かんすう}から数値^{りよう}を入力^{すうち}します。
数値^{にゅうりよく}を入力^{すうち}するときに、「input 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^{ふあぶん}するように、for文^{つか}とif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。
(入力^{にゅうりよく}した数値^{すうち}の文字数^{もじすう}だけ、文字列^{もじれつ}「● ▲ ■ ●▲■…」を表示^{ひようじ}します。)

【実行例】

```
>>> (executing file "for3-02.py")
input number=>2
●▲

>>> (executing file "for3-02.py")
input number=>5
●▲■●▲

>>> (executing file "for3-02.py")
input number=>9
●▲■●▲■●▲■

>>> (executing file "for3-02.py")
input number=>12
●▲■●▲■●▲■●▲■
```

3 次の仕様に基^{つぎ}づいて、プログラム^{しよ}を作成^{もとづい}してください。(ファイル名^{ぶろくらむ} : for3-03)

- ① input()関数^{いんぷつと}を利用して、キーボード^{かんすう}から数値^{りよう}を入力^{すうち}します。
数値^{にゅうりよく}を入力^{すうち}するときに、「input 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^{ふあぶん}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。
for2-04.pyのプログラム^{かんすう}を、range関数^{だい}の第3引数^{ひきすう}(ステップ)を使^{つか}わずに動作^{どうさ}するようにプログラム^かを変^かえてください。また、if文^{ぶん}を使^{つか}ってください。
(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^{ひようじ}します。)

【実行例】

```
>>> (executing file "for3-03.py")
input number=>5
5
8
11
14
17
20
23
26
29

>>> (executing file "for3-03.py")
input number=>9
9
12
15
18
21
24
27
30
```

4 次の仕様に基^{つぎ}づいて、プログラムを作成^{しようちょう}してください。(ファイル名^な : for3-04)

- ① input^{いんぷット}()関^{かん}数^{すう}を利用して、^{りよう}「正^{せい}の整^{せい}数^{すう}」を^{にゅうりよく}入^い力^{りき}します。
数^{すう}値^ちを^{にゅうりよく}入^い力^{りき}するときに、「input positive number=>」とメッセー^{ひようじ}ジ^じを^{ひようじ}表^{ひょう}示^じし
ます。
- ② 実行例^{じっこうれい}のとおり^どに^{どう}動^{どう}作^さする^{よう}に、for文^{ふおーぶん}を使^{つか}い^つく^くプログラム^つを^つ作^{つく}って^くだ^さい。
(0から^{にゅうりよく}入^い力^{りき}した^{すう}数^ち値^じま^{ひようじ}で^{ひようじ}表^{ひょう}示^じさせ^ます。このとき、偶^{ぐう}数^{すう}の^{ぐうすう}と^{ぐうすう}な^りに^は「even」、
奇^き数^{すう}の^{きすう}と^{きすう}な^りに^は「odd」と^{ひようじ}表^{ひょう}示^じし^ます。0より^{ちい}小^{すう}さい^{にゅうりよく}数^じ字^{ばあい}を^{にゅうりよく}入^い力^{りき}した^{ばあい}場^{ばあい}合^あい
は、「invalid!」と^{ひようじ}表^{ひょう}示^じし^ます。)

【実行例^{じっこうれい}】

```
>>> (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^{しよう}グ^{もとづい}ラム^{ふるくらむ}を作^{さくせい}成^ふして^あく^いだ^るさ^{めい}い^{ふおー}。 (ファイル名^ひ : for3-05)

- ① input()関^{いんぷつと}数^{かんすう}を^{りよう}利用^ふして、「負^ふの整^{せい}数^{すう}」を^{にゅうりよく}入^{すう}力^ちし^{にゅうりよく}ます。
数^{すう}値^ちを^{にゅうりよく}入^{すう}力^ちする^{にゅうりよく}と^{にゅうりよく}きに、「input negative 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^{じつ}行^{こう}例^{れい}の^{じつ}と^{こう}お^{れい}りに^{じつ}動^{どう}作^さする^{じつ}よ^さう^{じつ}に、「for文^{ふおーぶん}」を^{つか}使^{つか}い^{つか}プロ^{つか}グ^{つか}ラム^{つか}を^{つか}作^{つか}つ^{つか}て^{つか}く^{つか}だ^{つか}さ^{つか}い^{つか}。
- (-1から入^{にゅうりよく}力^{すう}した^{ひようじ}数^{ひようじ}値^じま^{ひようじ}で^{ひようじ}表^{ひようじ}示^{ひようじ}さ^{ひようじ}せ^{ひようじ}ま^{ひようじ}す。
- この^{ばいすう}と^{しかく}き、「2の^{ばいすう}倍^{しかく}数^{ばいすう}の^{しかく}と^{しかく}なり^{しかく}には^{しかく}「◆」^{しかく}、3の^{ばいすう}倍^{しかく}数^{ばいすう}の^{しかく}と^{しかく}なり^{しかく}には^{しかく}「◇」^{しかく}、6の^{ばいすう}倍^{しかく}数^{ばいすう}の^{しかく}と^{しかく}なり^{しかく}には^{しかく}「◆◇」^{しかく}、と^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ま^{ひようじ}す。0以^い上^{じょう}の^{すう}数^じを^{にゅうりよく}入^{すう}力^ちした^{にゅうりよく}場^ば合^{あい}は、「invalid!」と^{ひようじ}表^{ひようじ}示^{ひようじ}し^{ひようじ}ま^{ひようじ}す。)

【^{じつ}行^{こう}例^{れい}】

```
>>> (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^{ぎやうすう}」を入力^{にゆうりよく}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^{りやう}、行^{ぎやう}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^し表示^{ひやうじ}させるように^いします。い^いずれ^ずも0^{すうち}以下^{にゆうりよく}の数^{ばあい}値^じを^い入^い力^{りき}した場^{ばあい}合^じは、「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、プログラムを作成^{しやうせつ}してください。(ファイル名: for3-07)

- ① input^{いんぷつと}()関数^{かんすう}を利用して、キーボードから「行数^{ぎやうすう}」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、「row number =>」とメッセージを表示^{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。
(入力^{にゅうりよく}した行数^{ぎやうすう}分^{ぶん}、「▲」と「△」で作^{つく}った三角形^{さんかくけい}を表示^{ひやうじ}します。ただしこのとき、1行目^{ぎやうめ}は「▲」を1つ、2行目^{さんかく}は「△」を2つ、3行目^{さんかく}は「▲」を3つ…というように表示^{ひやうじ}します。また、行数^{ぎやうすう}に0以下^{いか}の数値^{すうち}を入力^{にゅうりよく}した場合は、「invalid!」と表示^{ひやうじ}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
▲
△△
▲▲▲
△△△△

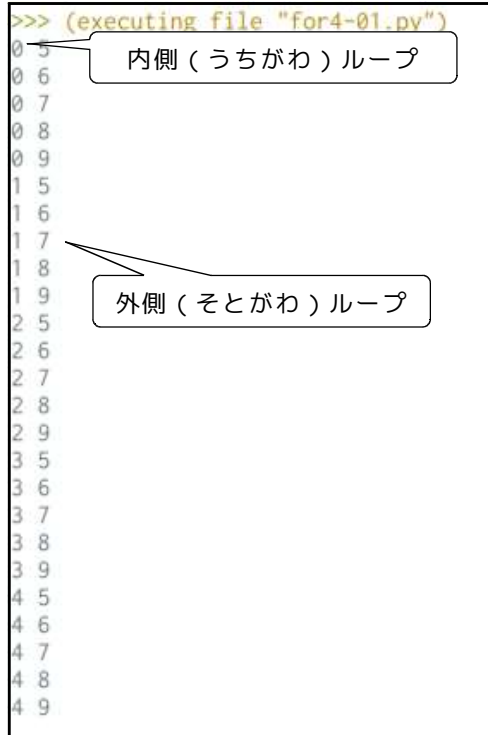
>>> (executing file "for3-07.py")
row number=>-1
invalid!
```

繰り返しfor文の復習問題④

- 1 次の仕様に基づいて、プログラムを作成してください。(ファイル名: for4-01)
- ① 実行例のとおり動作するように、for文の二重ループを使ったプログラムを作ってください。

【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}づいて、^{しよ}プログラムを^も作成^{とづい}してください。^{ふる}（^{くら}ファイル名^む：for4-02）^{さく}せいで

- ① 実行例^ふのとおり^あに動作^いするように、for文^るを使い^{めい}プログラム^ふを作^おってください。
（for4-01.pyを、for文^ぶの2重^んループ^じを使った^つプログラム^かを使って^つ書きかえます。
出力^しされる^ゆ数値^つのうち^く、3または5^すが出る^ち列^では「***」と^れ表示^つします）

【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^なされたときに、「invalid!」と表示^な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^{しよう}グラ^{もとづい}ムを^{ふる}作成^{くら}して^むください。^{さくせい} (ファイル名^ふ : for4-04^あ^い^る^{めい}^ふ^お)

- ① input()関^{いんぷつと}数^{かんすう}を^{りよう}利用^{すう}して、キー^{すう}ボ^ちード^にから^{ゆりよく}数^{すう}値^ちを^に入^に力^{ゆりよく}します。
数^{すう}値^ちを^に入^に力^{ゆりよく}する^{ひようじ}と^きに、「input number=>」と^{ひようじ}メ^せッ^じセ^じージ^じを^{ひようじ}表^じ示^じし^ます。
- ② 実行^{じっこうれい}例^{れい}の^{どう}と^おり^にに^{どう}動^さ作^さする^{よう}に、for^ふ文^{ぶん}を^{つか}使^{つか}い^てプロ^ふグラ^{ぶん}ム^{つか}を^{つか}作^{つか}っ^てく^ださ^い。
(for^ふ文^{ぶん}の^{じゅう}2^{じゅう}重^{じゅう}ル^{じゅう}ープ^{じゅう}を^{つか}使^{つか}った^{よう}に^にプロ^にグラ^にム^{すう}で^{つか}作^{すう}っ^てく^ださ^い。入^に力^{ゆりよく}した^{すう}数^ち値^じを^{つか}使^{つか}い、^{さん}三^{さん}角^{かく}形^{けい}を^いつ^くり^ます。た^だし、^い1^{いち}から^{すう}9^く以^に外^{ゆりよく}の^{すう}数^ち値^じが^に入^に力^{ゆりよく}さ^れた^とき^は、「i
nput only 1 - 9」^{ひようじ}と^{ひようじ}表^じ示^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^{しよつ}してください。(ファイル名^{もつ} : for4-06)

- ① input^{いんぷつと}()関数^{かんすう}を利用して^{りよう}、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^{ひようじ}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^{つか}の2重ループ^{じゅう}を使^{つか}ったプログラム^{つか}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}の文字数^{もじすう}だけ、文字列^{もじれつ}「1234512345…」^{ひようじ}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。)

【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

>>> (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}づいて、プログラムを作成^{しよつ}してください。(ファイル名: for4-07)

① input()関^{かん}数^{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}に動作^{どうさ}するように、for文^{ふーぶん}を使いプログラム^{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^{るか}を使^{つか}ったプログラム^{つか}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、文字列^{もじれつ}「○○●●○○…」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。)

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

【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
○●○○●
●○○●●
○●○○●
○●○○●
```