

# レポート

qitoy しゅどぼ

## 問 1

```
1 fn main() {
2     let s = {
3         let mut s = String::new();
4         std::io::stdin().read_line(&mut s).unwrap();
5         s.trim_end().to_owned()
6     };
7     let n: i32 = s.parse().unwrap();
8     println!(
9         "{}",
10        if n % 3 == 0 || s.contains('3') {
11            "aho".to_owned()
12        } else {
13            s
14        }
15    );
16 }
```

```
$ rustc main.rs
$ ./main
1
1

$ ./main
6
aho
```

```
$ ./main
```

```
13
```

```
aho
```

```
$ ./main
```

```
17
```

```
17
```

3 の倍数と 3 が付く数字のときは **aho** を返し、そうでないときは数字をそのまま返す世界のナベアツ [1] プログラムである。

$$[0, \text{len}_1) + [\text{len}_1, \text{len}_2) = [0, \text{len}_2)$$

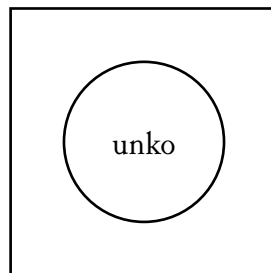


図 1

[1] 桂三度, Wikipedia

## 問 2

せっかくなのでこのソースコードを貼ってみる。

```
1 @require: azmath/azmath
2 @require: figbox/figbox
3 @require: code-printer/code-printer
4 @require: code-printer/code-theme
5 @require: code-printer/code-syntax
6 @require: code-printer/code-design
7 @require: bibyfi/bibyfi
8 @require: class-exdesign/exdesign
9 @require: class-exdesign/article-ja
```

```

10
11 let bib-default =
12   (|
13     title = ``;
14     author = None;
15     organization = None;
16     address = None;
17     edition = None;
18     month = None;
19     year = None;
20     note = None;
21     key = None;
22   |)
23 let bibs =
24   [
25     (`aho`, Manual(|
26       bib-default with title = `桂三度`;
27       author = Some([`Wikipedia`]);
28       address = Some(`https://ja.wikipedia.org/wiki/
29         %E6%A1%82%E4%B8%89%E5%BA%A6`);
30     |));
31 let mk-index ctx index =
32   let s = `[` ^ arabic index ^ `]`# in read-inline
33     ctx (embed-string s)
34 let mk-manual r =
35   match (r#author, r#address) with
36     | (Some([author]), Some(address)) -> let inner
37       = embed-string (r#title ^ `,`# ^ author) in { \
38         href (address) (inner); }
39     | _ -> { invalid }
40 let bibyfi-theme ctx index bib-item =
41   match bib-item with
42     | Manual(r) -> BiByFi.make-entry ctx (mk-index c
43       tx index) (read-inline ctx (mk-manual r))
44     | _ -> BiByFi.make-entry ctx (mk-index ctx index
45       ) (read-inline ctx { invalid })

```

```

41 let source =
42   CodePrinter.default
43   |> CodePrinter.set-syntax CodeSyntax.rust
44   |> CodePrinter.set-theme CodeTheme.basic-light
45   |> CodePrinter.set-line-break-mark (fun _ _ -> (
      inline-nil, inline-nil))
46 let prompt =
47   CodePrinter.default
48   |> CodePrinter.set-line-break-mark (fun _ _ -> (
      inline-nil, inline-nil))
49   |> CodePrinter.set-number-fun CodeDesign.number-
      fun-null
50 let-math \range m1 m2 = math-paren AZMathParens.squa
      re-bracket-l AZMathParens.round-bracket-r  $\{ \#m1 , \#m
      2 \}$ 
51 let-math \len = math-char MathOrd `len`
52 let circle-text ((x, y) as c) r it ctx =
53   let ib = read-inline ctx it in
54   let gr = draw-text c ib in
55   let ((xmin, ymin), (xmax, ymax)) = get-graphics-b
      box gr in
56   let (xwid, ywid) = (xmax -' xmin, ymax -' ymin) i
      n
57   let text = shift-graphics (0pt -' xwid *' 0.5, 0pt
      -' ywid *' 0.5) gr in
58   [
59     Gr.circle c r
60     |> stroke 1pt Color.black;
61     text;
62   ]
63 in
64
65 document(|
66   title = { レポート };
67   author = { qitoy しゅどぼ };
68   date = {};
69   show-title = true;

```

```

70 show-toc = false;
71 style = ArticleJa.a4paper;
72 design = (|
73   ArticleJa.article with section-num-function = fu
74     n _ i -> `問` ^ arabic i;
75   |);
76 header-footer = ArticleJa.normalHF;
77 fonts = ArticleJa.fonts;
78 |)'<
79 +section {} <
80   +file-printer ?:(source) (`main.rs`);
81   +code-printer ?:(prompt) (`$ rustc main.rs
82 $ ./main
83 1
84 1
85 $ ./main
86 6
87 aho
88
89 $ ./main
90 13
91 aho
92
93 $ ./main
94 17
95 17`);
96
97 +p { 3の倍数と3が付く数字のときは\inline-code (`aho`);
98   を返し、そうでないときは数字をそのまま返す世界のナビアツ\cite [
99   `aho`];};プログラムである。 }
100
101 +eqn (${ \range{0}{\len_{1}} + \range{\len_{1}}{
102   \len_{2}} = \range{0}{\len_{2} \notag}});

```

```
103     from-graphics (100pt, 100pt) []
104     |> graffiti-given-context (
105         circle-text (50pt, 50pt) 30pt {unko}
106     ) |> frame 1pt Color.black;
107     gap 10pt;
108     textbox {図1};
109     ]));
110 >
111
112 +makebibliography (bibyfi-theme) (bibs);
113
114 +section {} <
115     +p { せっかくなのでこのソースコードを貼ってみる。 }
116     +file-printer ?:(source
117         |> CodePrinter.set-syntax CodeSyntax.satysfi)
118         (`hoge.saty`);
119 >
```