レポート

qitoy しゅどぼ

問 1

```rust
fn main() {
    let s = {
        let mut s = String::new();
        std::io::stdin().read_line(&mut s).unwrap();
        s.trim_end().to_owned()
    };
    let n: i32 = s.parse().unwrap();
    println!(
        "{}",
        if n % 3 == 0 || s.contains('3') {
            "aho".to_owned()
        } else {
            s
        }
    );
}
```

```
$ 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。

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

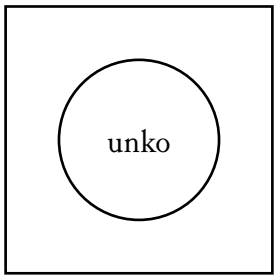

図 1

[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。

```
@require: azmath/azmath
@require: figbox/figbox
@require: code-printer/code-printer
@require: code-printer/code-theme
@require: code-printer/code-syntax
@require: code-printer/code-design
@require: bibyfi/bibyfi
@require: class-exdesign/exdesign
@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/
          %E6%A1%82%E4%B8%89%E5%BA%A6`);
29      |));
30    ]
31  let mk-index ctx index =
32    let s = `[` ^ arabic index ^ `] `# in read-inline
      ctx (embed-string s)
33  let mk-manual r =
34    match (r#author, r#address) with
35      | (Some([author]), Some(address)) -> let inner
        = embed-string (r#title ^ `, `# ^ author) in { \
  href (address) (inner); }
36      | _ -> { invalid }
37  let bibyfi-theme ctx index bib-item =
38    match bib-item with
39      | Manual(r) -> BiByFi.make-entry ctx (mk-index c
  tx index) (read-inline ctx (mk-manual r))
40      | _ -> BiByFi.make-entry ctx (mk-index ctx index
      ) (read-inline ctx { invalid })
```

```
let source =
  CodePrinter.default
    |> CodePrinter.set-syntax CodeSyntax.rust
    |> CodePrinter.set-theme CodeTheme.basic-light
    |> CodePrinter.set-line-break-mark (fun _ _ -> (inline-nil, inline-nil))
let prompt =
  CodePrinter.default
    |> CodePrinter.set-line-break-mark (fun _ _ -> (inline-nil, inline-nil))
    |> CodePrinter.set-number-fun CodeDesign.number-fun-null
let-math \range m1 m2 = math-paren AZMathParens.square-bracket-l AZMathParens.round-bracket-r ${#m1 , #m2}
let-math \len = math-char MathOrd `len`
let circle-text ((x, y) as c) r it ctx =
  let ib = read-inline ctx it in
  let gr = draw-text c ib in
  let ((xmin, ymin), (xmax, ymax)) = get-graphics-bbox gr in
  let (xwid, ywid) = (xmax -' xmin, ymax -' ymin) in
  let text = shift-graphics (0pt -' xwid *' 0.5, 0pt -' ywid *' 0.5) gr in
    [
      Gr.circle c r
        |> stroke 1pt Color.black;
      text;
    ]
in

document(|
  title = { レポート };
  author = { qitoy しゅどぼ };
  date = {};
  show-title = true;
```

```
70    show-toc = false;
71    style = ArticleJa.a4paper;
72    design = (|
73      ArticleJa.article with section-num-function = fu
      n _ i -> `問` ^ arabic i;
74    |);
75    header-footer = ArticleJa.normalHF;
76    fonts = ArticleJa.fonts;
77  |)'<
78    +section {} <
79      +file-printer ?:(source) (`main.rs`);
80      +code-printer ?:(prompt) (`$ rustc main.rs
81  $ ./main
82  1
83  1
84
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`);
      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\cite [
      `aho`;];プログラムである。 }
98
99      +eqn (${\range{0}{\len_{1}} + \range{\len_{1}}{
      \len_{2}} = \range{0}{\len_{2} \notag}});
100
101     +fig-center (FigBox.(
102       vconcat ?:align-center [
```

```
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)
          (`hoge.saty`);
118     >
119 >
```