

# 壁パズル大会 2020 解答

開成学園パズル研究部

2020/9/20

## 1 同じ数探し

「去年は3階」と「次の部長も会計も中3」より、「3」が答えです。

## 2 規則性

1つめの規則性は、「千円札に描かれている人物の最初の文字」です。2024年から北里柴三郎になることが発表されていますね。

1. 日本武尊
2. 聖徳太子
3. 伊藤博文
4. 夏目漱石
5. 野口英世
6. 北里柴三郎

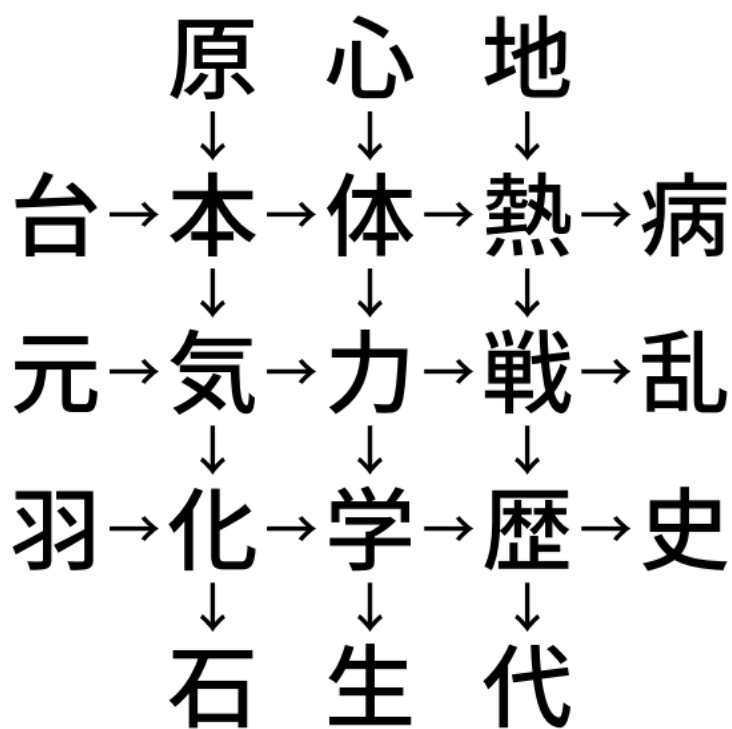
より、答えは「野」です。

2つめの規則性は「0以上の整数を漢字で書いた時の画数」です。0を「零」と書いたので13となっています。

「8」の漢字の画数なので、答えは「2」です。

### 3 漢字パズル

#### 3.1 和同開珎

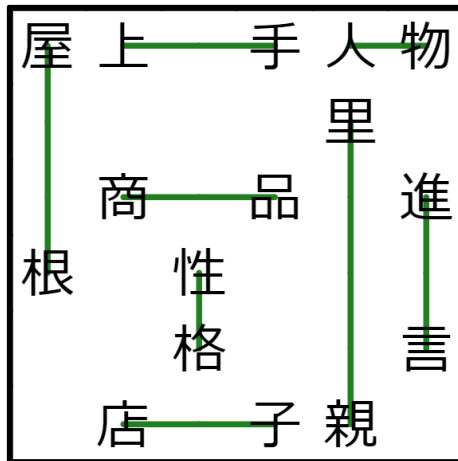


和同開珎回答

と解けるので、答えは「本・体・熱・氣・力・戦・化・学・歴」でした（文字の区切り方は自由）。

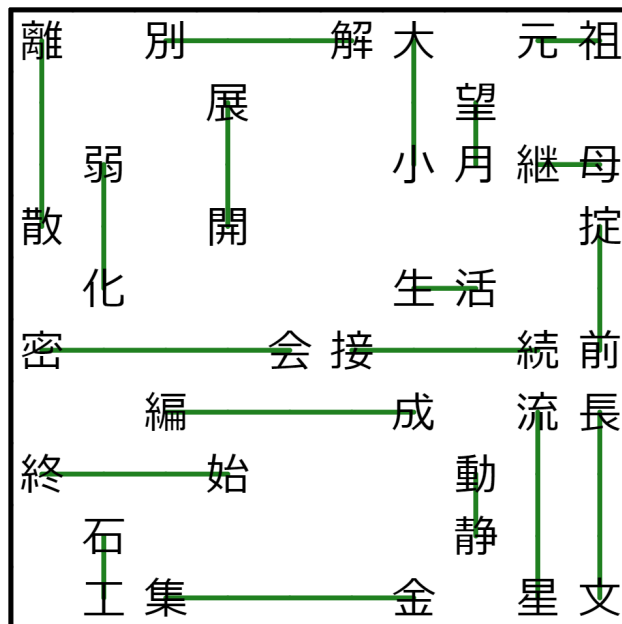
また、「本・筋・合・氣・力・戦・化・学・歴」と「本・地・熱・氣・力・戦・化・学・歴」の解も存在しました。複数解が出てしまい、申し訳ありませんでした。

### 3.2 漢字リンク



漢字リンク 1 問目回答

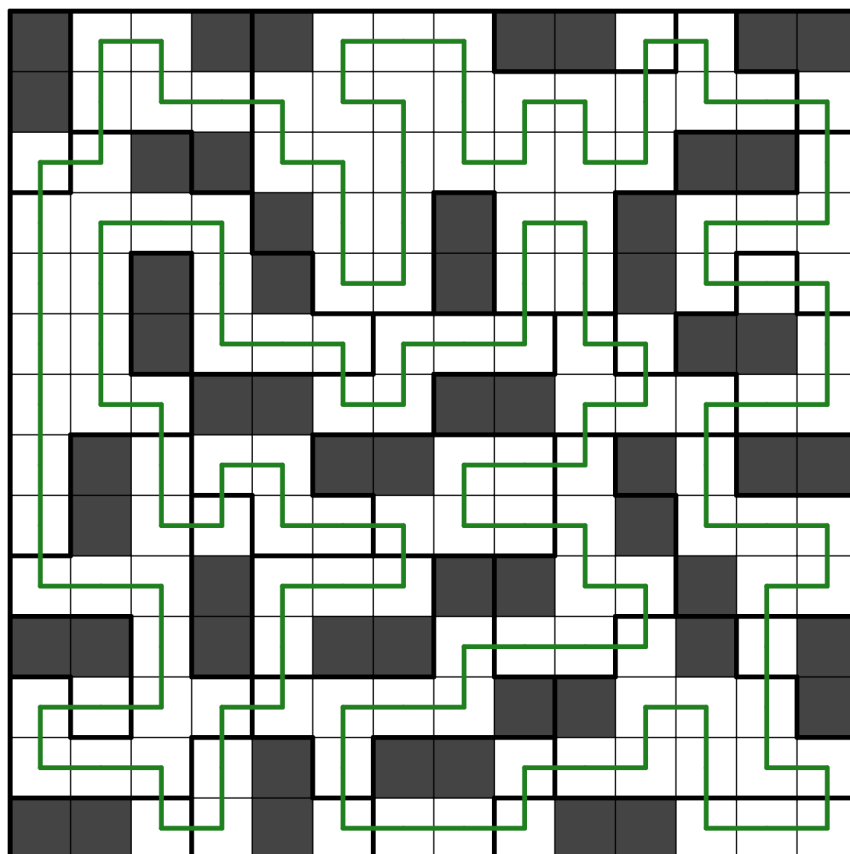
と解けるので、答えは「16」です。



漢字リンク 2 問目回答

と解けるので、答えは「90」です。

## 4 のりのるーぷ



のりのるーぷ回答

と解けるので、答えは「12」です。

## 5 論理パズル

- a 君「和が 19 以上の素数」
- b 君「両方とも偶数」
- c 君「2 枚とも書かれているカードはアルファベット」
- d 君「差が 9 以上」
- e 君「2 枚とも素数で、和も素数」
- f 君「和が 8 以下」

という条件から、

- a: (6, 13), (7, 12), (8, 11), (9, 10), (10, 13), (11, 12)

- b: (4, 6), (4, 8), (4, 10), (4, 12), (6, 8), (6, 10), (6, 12), (8, 10), (8, 12), (10, 12)
- c: (1, 11), (1, 12), (1, 13), (11, 12), (11, 13), (12, 13)
- d: (1, 10), (1, 11), (1, 12), (1, 13), (3, 12), (3, 13), (4, 13)
- e: (2, 3), (2, 5), (2, 11)
- f: (1, 3), (1, 4), (1, 5), (1, 6), (3, 4)

までは絞り込めます。

あとは成立する組み合わせをすべて探し出すだけです。単純な組み合わせ数は5万を超えますが、「数字が一切重複しない」という性質を利用するとかなり絞り込めます。

実際に全て調べ上げると、

1. a: (7, 12), b: (6, 8), c: (11, 13), d: (1, 10), e: (2, 5), f: (3, 4)
2. a: (9, 10), b: (4, 6), c: (11, 12), d: (3, 13), e: (2, 5), f: (1, 7)
3. a: (9, 10), b: (4, 6), c: (11, 13), d: (3, 12), e: (2, 5), f: (1, 7)
4. a: (9, 10), b: (4, 8), c: (11, 12), d: (3, 13), e: (2, 5), f: (1, 6)
5. a: (9, 10), b: (4, 8), c: (11, 12), d: (3, 13), e: (2, 5), f: (1, 7)
6. a: (9, 10), b: (4, 8), c: (11, 13), d: (3, 12), e: (2, 5), f: (1, 6)
7. a: (9, 10), b: (4, 8), c: (11, 13), d: (3, 12), e: (2, 5), f: (1, 7)
8. a: (9, 10), b: (6, 8), c: (1, 12), d: (4, 13), e: (2, 11), f: (3, 5)
9. a: (9, 10), b: (6, 8), c: (11, 12), d: (1, 13), e: (2, 5), f: (3, 4)
10. a: (9, 10), b: (6, 8), c: (11, 12), d: (3, 13), e: (2, 5), f: (1, 4)
11. a: (9, 10), b: (6, 8), c: (11, 12), d: (3, 13), e: (2, 5), f: (1, 7)
12. a: (9, 10), b: (6, 8), c: (11, 12), d: (4, 13), e: (2, 3), f: (1, 5)
13. a: (9, 10), b: (6, 8), c: (11, 12), d: (4, 13), e: (2, 3), f: (1, 7)
14. a: (9, 10), b: (6, 8), c: (11, 12), d: (4, 13), e: (2, 5), f: (1, 3)
15. a: (9, 10), b: (6, 8), c: (11, 12), d: (4, 13), e: (2, 5), f: (1, 7)
16. a: (9, 10), b: (6, 8), c: (11, 13), d: (1, 12), e: (2, 5), f: (3, 4)
17. a: (9, 10), b: (6, 8), c: (11, 13), d: (3, 12), e: (2, 5), f: (1, 4)
18. a: (9, 10), b: (6, 8), c: (11, 13), d: (3, 12), e: (2, 5), f: (1, 7)
19. a: (9, 10), b: (6, 8), c: (12, 13), d: (1, 11), e: (2, 5), f: (3, 4)

と、19通りできます。

登場人物は全員賢いので、全員の頭の中にこの19通りの表が存在したはずです。

さて、「3人が『全員のカードがわかった』と同時に宣言した」ことから、「自分の持っているカードの組み合わせがこの19通りの中で重複していない」という人が3人いたことがわかります。

カードの組み合わせが他と重複していない人の数を調べると、1番目はaとdの2人・8番目はcとeとfの3人・9番目はdの1人・12番目はfの1人・14番目はfの1人・16番目はdの1人・19番目はcとdの2人となっています。

よって、3人いる8番目の

- a: (9, 10)
- b: (6, 8)

- c: (1, 12)
- d: (4, 13)
- e: (2, 11)
- f: (3, 5)

が答えとわかります。

上の調べ上げはプログラムを組むことで省力化することができます。実際に上の調べ上げを行う Rust<sup>\*1</sup>のプログラムは以下のようになります<sup>\*2</sup>。

```

1 // カードの組み合わせ
2 #[derive(Debug)]
3 struct Card {
4     a: (u32, u32),
5     b: (u32, u32),
6     c: (u32, u32),
7     d: (u32, u32),
8     e: (u32, u32),
9     f: (u32, u32),
10 }
11
12 // 重複が無いかを確認する
13 fn check(card: &Card) -> bool {
14     let (a1, a2) = card.a;
15     let (b1, b2) = card.b;
16     let (c1, c2) = card.c;
17     let (d1, d2) = card.d;
18     let (e1, e2) = card.e;
19     let (f1, f2) = card.f;
20     let old_vec = vec![a1, a2, b1, b2, c1, c2, d1, d2, e1, e2, f1, f2];
21     let mut new_vec = old_vec.clone();
22     new_vec.sort();
23     new_vec.dedup();
24     old_vec.len() == new_vec.len()
25 }
26
27 fn main() {
28     // aからfのあり得る組み合わせのリスト
29     let a_lst = vec![(6, 13), (7, 12), (8, 11), (9, 10), (10, 13), (11, 12)];

```

<sup>\*1</sup> <https://www.rust-lang.org/>

<sup>\*2</sup> このコードを L<sup>A</sup>T<sub>E</sub>X で組むのに、Tobias Denkinge 氏によって作成された listings-rust パッケージ (<https://github.com/denki/listings-rust>) を使用させていただきました

```

30 let b_lst =
31     vec![
32         (4, 6), (4, 8), (4, 10), (4, 12), (6, 8),
33         (6, 10), (6, 12), (8, 10), (8, 12), (10, 12)
34     ];
35 let c_lst = vec![(1, 11), (1, 12), (1, 13), (11, 12), (11, 13), (12, 13)];
36 let d_lst =
37     vec![
38         (1, 10), (1, 11), (1, 12), (1, 13),
39         (3, 12), (3, 13), (4, 13)
40     ];
41 let e_lst = vec![(2, 3), (2, 5), (2, 11)];
42 let f_lst = vec![(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5)];
43 // 全組み合わせを作成する
44 let mut card_lst = Vec::new();
45 for a in &a_lst {
46     for b in &b_lst {
47         for c in &c_lst {
48             for d in &d_lst {
49                 for e in &e_lst {
50                     for f in &f_lst {
51                         card_lst.push(Card {
52                             a: *a,
53                             b: *b,
54                             c: *c,
55                             d: *d,
56                             e: *e,
57                             f: *f,
58                         })
59                     }
60                 }
61             }
62         }
63     }
64 }
65
66 // 重複が無いカードの組み合わせを選び出す
67 let card_lst: Vec<&Card> = card_lst.iter().filter(|card| check(card)).collect();
68
69 // 表示する

```

```
70 println!("{:?}", card_lst);  
71 }
```