模試解説(if)

3年情報

問１　ア・イ

●勅令の１は４で割って割り切れる場合は閏年

▼

●amariが４で割って０になるということなので

答え　ア　①
　　　イ　③

問１　ウ

●表示のルールは文字は” ”

●変数はそのまま

●文字と変数をつなぐときは,

答え　ウ　０

問２　エ

●勅令の２の条件は６６０を引いた年で１００で割りきれる。

答え　エ　②

問２　オ

- 勅令の２の条件は６６０を引いた年で１００で割りきれる。
- かつ１００で割ったときの商が４で割り切れない

答え　オ　③

問２　カ

答え　オ　③

閏年の条件をまとめると

1. 4で割り切れる年：基本的に、年が4で割り切れる場合は閏年の候補になります。
    1. 例：2020年、2024年など。
2. 100で割り切れる年：ただし、その年が100で割り切れる場合は、閏年ではありません。
    1. 例：1900年、2100年などは閏年ではない。
3. 400で割り切れる年：100で割り切れる年の中でも、その年が400で割り切れる場合は閏年になります。
    1. 例：1600年、2000年などは閏年。

●まとめ

- 閏年：4で割り切れるが、100で割り切れない年、または400で割り切れる年。
- 閏年ではない：100で割り切れるが400で割り切れない年。

閏年の条件をまとめると

●閏年：4で割り切れるが、100で割り切れない年、または400で割り切れる年。

答え　キ　0
　　　ク　③
　　　ケ　④

答え　コ　③
　　　サ　④
　　　シ　⑤

10進数→2進数の変換

① 29

2) 29
2) 14 ・・・ 1
2) 7 ・・・ 0
2) 3 ・・・ 1
2) 1 ・・・ 1
　　 0 ・・・ 1

答え

$11101_{(2)}$

16進数について

0 1 2 3 4 5 6 7 8 9 A B C D E F　の16個の数を使用。

16　倍ずつ位（くらい）が上がる。

- 「9」の次は「10」ではなく、「A」を用いる
- 1つのケタの最大の数「F」の次にケタが上がり、「10」となる

１０進数　　　　１６進数

16進数→10進数の変換

①A3

桁の重み （ 16^1 ）（ 16^0 ）

×　×

A　3

▼

$16^1 \times 10 + 16^0 \times 3$

答え　163

16進数→10進数の変換

①173

16) 173

16) 10 ・・・ 13 D

0 ・・・ 10 A

答え

AD (16)

2進数→16進数

	上位4ビット	下位4ビット
2進数	0101	1110
	5	14
16進数	5	E

- ポイントは4ビットずつ分ける

次の中でコンピューターが計算できるものはどれ?

①3+6　②4×3　③7-2　④10÷5

①３＋６のみ
コンピュータは足し算しかできない

- 引き算や掛け算を全て足し算になおしてから計算しています。
- コンピュータの計算は論理回路の組み合わせで実現されています。
  論理回路を複雑にすると計算スピードが落ちるので、究極のシンプルな形を追い求めこうなりました

補数について

●補数とは・・・元の数を足したときに桁上がりする最小の数

のことを指しています

例）10進法における4に対する10の補数は6、

23に対する10の補数は77

補数で計算すると

①7 - 2

2に対する
補数は8

10進法：7-2=7＋8＝~~1~~5
補数

最上位である1(桁上がり部分)
を取り除き5

①5 - 3

3に対する
10の補数は7

足し算を使った式：5＋7＝12

最上位である1を取り除き2

②8 - 4

4に対する
10の補数は6

足し算を使った式：8+4=14

↓

最上位である1を取り除き4

2進数の足し算


```
    1  1  1
    0  1  1  1
+)  0  1  0  1
--------------
 1  1  0  0
```

☆ポイント

1+1=10

補数の求め方について

例）７－２

10進法：７－２＝７＋８＝１５
補数

10進法で2の補数は８になります

10進法で補数を求める方法は10-2をすれば求まります。

２進法でも同じように引き算をすれば求まります。

７－２を２進法にすると

２進法：$0111_{(2)} - 0010_{(2)}$

２進法で補数を求める方法は$10000_{(2)} - 0010_{(2)}$をすれば求まります。

でもこれっておかしくないですか？

2の補数の求め方

例） ０１０１の場合

０００１

１１１０

１１１１

手順①
ビットを反転
（１の補数）

手順②
１を加算

この場合
下位４ビットだけとる（桁上がりは無視）

①００１１の場合

００１１$_{(2)}$

手順①
ビットを反転
（１の補数）

１１００

手順②
１を加算

１１０１

②00111000の場合

0 0 1 1 1 0 0 0 $_{(2)}$

1 1 0 0 0 1 1 1

1 1 0 0 1 0 0 0

手順①
ビットを反転
（1の補数）

手順②
1を加算

２の補数表現使った足し算で求める方法

①$0100_{(2)} - 0011_{(2)}$

手順①
右側の２進法の補数を求める

0011の補数を求める

●0011の場合

0 0 1 1 $_{(2)}$

1 1 0 0 $_{(2)}$

1 1 0 1 $_{(2)}$

①ビットを反転
（1の補数）

②1を加算

２の補数表現使った足し算で求める方法

①$0100_{(2)} - 0011_{(2)}$

手順①
右側の２進法の補数を求める

▼

$1101_{(2)}$

２の補数表現使った足し算で求める方法

$①0100_{(2)} + 1101_{(2)}$

$= 10001_{(2)}$

手順③
下位４ビットだけとる

$= 0001_{(2)}$

②$0111_{(2)} - \underline{0100}_{(2)}$

▼

補数にすると・・・

$1100_{(2)}$

②$0111_{(2)} + 1100_{(2)}$

$= 10011_{(2)}$

手順③

<u>下位４ビットだけとる</u>

$= 0011_{(2)}$

②$0110_{(2)} - \underline{0001}_{(2)}$

▼

補数にすると・・・

$1111_{(2)}$

②$0110_{(2)} + 1111_{(2)}$

$= 10101_{(2)}$

手順③
下位４ビットだけとる

$= 0101_{(2)}$

コンピュータでのマイナス表現

4 ビットでは・・・

1 番上位のビット（先頭のビット）が

0 のとき ⋯⋯→ ②正

1 のとき ⋯⋯→ ③負

4 ビットで表される

最大値 ⋯⋯→ 7

最小値 ⋯⋯→ −8

この先頭のビットを ①符号ビット

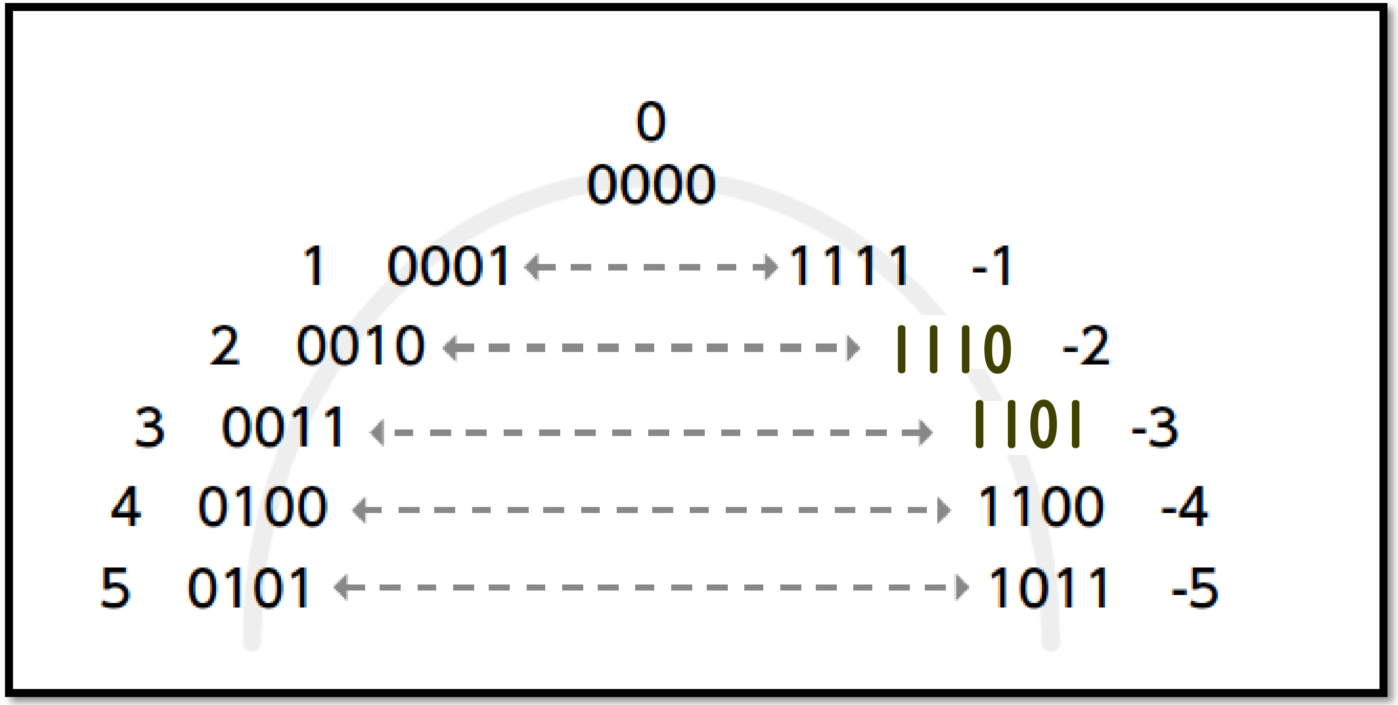
- 3ビットの場合もある

2進法表現 5	2進法表現 4321	2の補数表現での数値	符号なし整数での数値
1	0111	7	7
1	0110	6	6
1	0101	5	5
1	0100	4	4
1	0011	3	3
1	0010	2	2
1	0001	1	1
1	0000	0	0
0	1111	-1	15
0	1110	-2	14
0	1101	-3	13
0	1100	-4	12
0	1011	-5	11
0	1010	-6	10
0	1001	-7	9
0	1000	-8	8

先頭のビット

表５　整数の２の補数表現

なぜ符号ビットを使う？



１を補数変換をして

－１にするには

０と１を反転して

１を足す

「１１１０」の表現するとこれが「－２」か「１４」を表す数なのかわからない。そこで「符号付きビットで表現（2の補数表現）」のように断り書き付くことが多い。

全統模試

注意点

- 10進数から2進数に変換をして答えを出しても〇〇ビットでと指定が入る時がある
- 例　29を2進数に直すと11101

　　これは5ビット(1と0が5つ並んでいる)

これを7ビットで表現しなさいと言われるとどうするか？

▼

先頭に0を付け加えて7ビットにする

$0011101_{(2)}$

●1の2進数表記は？

$$2\,\overline{)\,1\,}$$

0 ・・・ 1

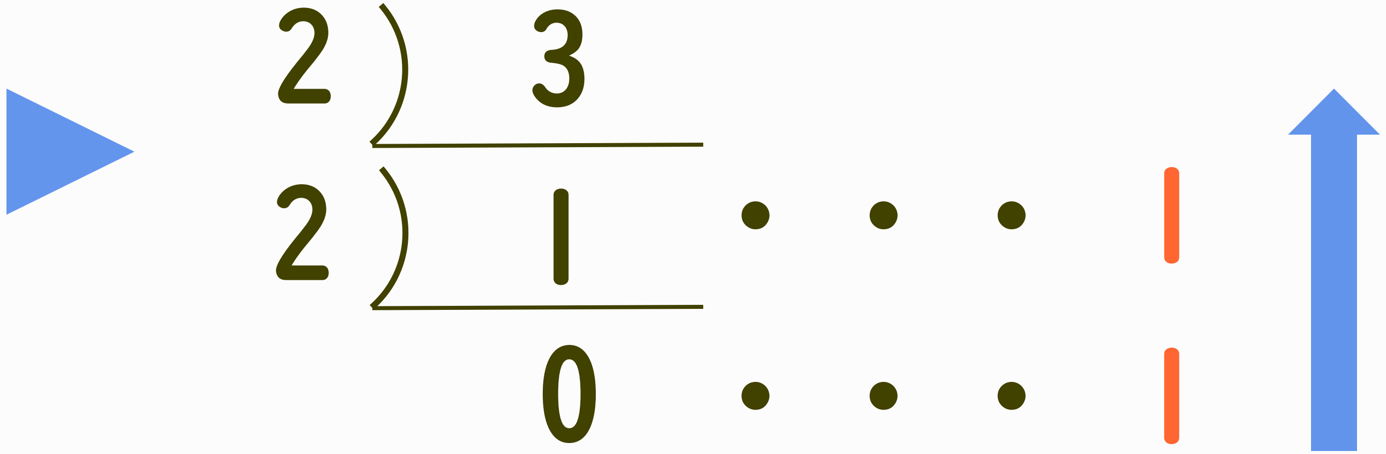
●答えは1だが、3ビット表されるとあるので

無理やり0を1の前に足して3ビットにする

答え ①001

全統模試 ｜ キ

●3の2進数表記は？



●カ同様に無理やり0を1の前に足して3ビットにする

答え ③ 011

●手順としてはまず2を3ビットの2進数にする

▶ 2) 2

2) 1 ・・・ 0

0 ・・・ 1

●カ、キ同様に無理やり0を1の前に足して3ビットにする

答え 010

●次に010（2）を-2にする方法を考える

▼

2を－１にするには

①０と１を反転

②１を足す

答え　110

● 110は符号ビットを使って－を表現している
ここで問われているのは符号ビットを使わずに
10進数に変換する

桁の重み（ 2^2 ）（ 2^1 ）（ 2^0 ）

×　×　×

1　1　0

▼

4 ＋ 2 ＋ 0

答え　6

●4ビット目と3ビット目がオンである。
コンピュータではオフを0、オンを1と表現する。

●別の解き方として
4ビット目と3ビット目が生きていて、それぞれの数字が8と4。
合計すると12である。これを2進数になおす。

答え　1100

問題 | オ

●まず1から16までの数字を思い浮かべてと指示している

●実際にカードにあるのは1から15までの数字

▼

●カードにならないとなる数字となると16が当てあまる

答え ②

問題　|　カ

●今回のカードゲームでは4枚のカードを用意している。
その上で1～16までの数字を当てる。
一番大きい数字は$2^4=16$と考える

●同じ考えで2^7で一番大きい数字を求めることができる

答え　⑤

問題　｜　キ

●まず32ビットあるものを8ビットずつ区切る。

▼

10101100　00010000　00001010　10110100

▼

●区切ったものを10進数に変換する。

▼

172.　16.　10.　180

答え　③

$$
\begin{array}{r}
1\quad\quad 1\quad\quad \\
11001 \\
+)\ 10101 \\
\hline
101110
\end{array}
$$

☆ポイント

$1+1=10$

答え　⑤

$11001_{(2)} - \underline{10101}_{(2)}$

手順①
右側の２進法の補数を求める

▼

$01011_{(2)}$

$11001_{(2)} - \underline{10101}_{(2)}$

手順①
右側の２進法の補数を求める

▼

$01011_{(2)}$

$$11001_{(2)} + 01011_{(2)}$$

$$= 100100_{(2)}$$

手順③
下位４ビットだけとる

$$= 00100_{(2)}$$

コ　補数とは・・・元の数を足したときに桁上がりする最小の数のことを指しています

- 10進法における4に対する10の補数は6、23に対する10の補数は77

答え　②

問題　｜　サ

●2の補数の求め方は

0 0 0 1

手順①
ビットを反転
（1の補数）

1 1 1 0

手順②
1を加算

1 1 1 1

答え　⑦

問題　|　シ

●10011100の補数の求め方は

10011100

手順①
ビットを反転
（1の補数）

01100011

手順②
1を加算

01100100

答え　④

●1011の補数の求め方は

1011

手順①
ビットを反転
（1の補数）

0100

手順②
1を加算

0101

答え　②

$$
\begin{array}{r}
1\,1\,1\,1 \\
0\,1\,0\,1 \\
+)\ 1\,1\,1\,1 \\
\hline
1\,0\,1\,0\,0
\end{array}
$$

答え ⑥

問題 | ソ

$$\begin{array}{r} 1\,1\,1\,1 \\ 0\,1\,0\,1 \\ +)\ 1\,1\,1\,1 \\ \hline \cancel{1}\,0\,1\,0\,0 \end{array}$$

答え ①