2025年度　総合政策学部　一般選抜　問題訂正

教科・科目	ページ	設問	誤	→	正
情報 および 数学	16	数学 V	1行目 「関数$f(x)$と$g(x)$」	→	「xの2次関数$f(x)$と 1次関数$g(x)$」

2025年度

慶應義塾大学入学試験問題

総合政策学部

情報および数学

注意事項1

1. 試験開始の合図があるまで、この問題冊子を開かないでください。
2. この冊子は全部で20ページです。
   - 情報の問題Ⅰ～Ⅲは4ページから12ページです。
   - 数学の問題Ⅳ～Ⅵは14ページから18ページです。
3. 試験開始の合図とともにすべてのページが揃っているか確認してください。
   ページの欠落・重複があった場合には、直ちに監督者に申し出てください。
4. 問題冊子の2ページに「注意事項2」があります。3ページに「注意事項3」があります。試験開始後必ず読んでください。
5. 問題冊子は、試験終了後必ず持ち帰ってください。
6. 受験番号と氏名は、解答用紙の所定の欄に必ず記入してください。
7. 解答用紙の「注意事項」を必ず読んでください。

注意事項2

問題冊子に数字の入った $\boxed{}$ があります．それらの数字は解答用紙の解答欄の番号を表しています．対応する番号の解答欄の 0 から 9 までの数字または - (マイナスの符号) をマークしてください．

$\boxed{}$ が 2 個以上つながったとき，数は右詰めで入れ，左の余った空欄には 0 を入れてください．負の数の場合には，マイナスの符号を先頭の $\boxed{}$ に入れてください．また，小数点以下がある場合には，左詰めで入れ，右の余った空欄には 0 を入れてください．

(例)　$12 \longrightarrow \boxed{0}\boxed{1}\boxed{2}$　　　$-3 \longrightarrow \boxed{-}\boxed{0}\boxed{3}$

$1.4 \longrightarrow \boxed{0}\boxed{0}\boxed{1}.\boxed{4}\boxed{0}$　　　$-5 \longrightarrow \boxed{-}\boxed{0}\boxed{5}.\boxed{0}\boxed{0}$

分数は約分した形で解答してください．マイナスの符号は分母には使えません．

(例)　$\dfrac{4}{8} \longrightarrow \dfrac{1}{2} \longrightarrow \dfrac{\boxed{0}\boxed{1}}{\boxed{0}\boxed{2}}$　　　$-\dfrac{6}{9} \longrightarrow -\dfrac{2}{3} \longrightarrow \dfrac{\boxed{-}\boxed{2}}{\boxed{0}\boxed{3}}$

ルート記号の中は平方因子を含まない形で解答してください．

(例)　$\sqrt{50} \longrightarrow \boxed{0}\boxed{5}\sqrt{\boxed{0}\boxed{2}}$　　　$-\sqrt{24} \longrightarrow \boxed{-}\boxed{2}\sqrt{\boxed{0}\boxed{6}}$

$\sqrt{13} \longrightarrow \boxed{0}\boxed{1}\sqrt{\boxed{1}\boxed{3}}$　　　$-\dfrac{\sqrt{18}}{6} \longrightarrow \dfrac{\boxed{-}\boxed{1}\sqrt{\boxed{0}\boxed{2}}}{\boxed{0}\boxed{2}}$

数式については，つぎの例のようにしてください．分数式は約分した形で解答してください．

(例)　$\sqrt{12a} \longrightarrow \boxed{0}\boxed{2}\sqrt{\boxed{0}\boxed{3}\,a}$

$-a^2-5 \longrightarrow \boxed{-}\boxed{1}\,a^2+\boxed{0}\boxed{0}\,a+\boxed{-}\boxed{5}$

$\dfrac{4a}{2a-2} \longrightarrow \dfrac{-2a}{1-a} \longrightarrow \dfrac{\boxed{0}\boxed{0}+\boxed{-}\boxed{2}\,a}{1-\boxed{0}\boxed{1}\,a}$

また，選択肢の番号を選ぶ問題では，最も適切な選択肢を 1 つだけ選んでください．同じ選択肢を複数回選んでもかまいません．

注意事項3

問題文中のプログラムの表記は、大学入学共通テストで使用されている「共通テスト用プログラム表記」に準じます。以下にその概要を示します。

文字列　「"SFC"」のようにダブルクォーテーションで囲みます。複数の文字列を「+」で結合できます。

配列　「[1, 2, 3]」のように、要素の並びを角括弧で囲みます。

変数　変数名は英字で始まる英数字とアンダースコアの並びです。ただし、配列変数名は先頭文字が大文字です。配列の要素は、「Data[0]」のように、配列変数名の後に角括弧で囲んだ添字を書きます。添字は0から始まります。

代入文　「count = 1」のように等号の左辺に変数、右辺に式を書きます。変数には、数値だけでなく文字列や配列も代入できます。

例：name = "Keio " + "SFC"

例：Data = [1, 2, 3]

例：Data[0] = 4

演算子　加減乗除と不等号は数学と同じです。数学と異なるのは次の演算子です。

整数の除算：商（整数）「÷」、余り「%」

比較演算：等しい「==」、等しくない「!=」、以上「>=」、以下「<=」

論理演算：論理積「and」、論理和「or」、否定「not」

関数　関数呼び出しは、「**平均**(Data)」のように、関数名の後に括弧で囲んだ引数を書きます。

「表示する」という関数は、引数の数値や文字列を画面に表示する関数です。それ以外の関数は問題文中に説明があります。

制御文　条件分岐や繰り返しの制御文の下にある罫線は、その制御文の制御範囲を表します。「└」が制御範囲の最後の文です。

例：
```
もし n % 2 == 0 ならば：
│ n = n ÷ 2
そうでなければ：
└ n = 3 * n + 1
```

例：
```
i を 0 から 2 まで 1 ずつ増やしながら繰り返す：
└ sum = sum + Data[i]
```

総

情報 I

データ伝送等で使われる誤り検出のための符号について述べた次の文章の空欄 $\boxed{(1)}$ ~ $\boxed{(14)}$、$\boxed{(23)}$ ~ $\boxed{(29)}$ に入るもっとも適した語を下の選択肢から選び，その番号を解答欄にマークしなさい。また、空欄 $\boxed{(15)}\boxed{(16)}$ ~ $\boxed{(20)}\boxed{(21)}\boxed{(22)}$ に入るもっとも適した数字を解答欄にマークしなさい。

データ伝送の際に 2 進法で表現されたデータの一部が誤って伝送された場合、それを検出したり訂正したりする仕組みが考えられており、誤り検出や誤り訂正と呼ばれている。単純なものとしては、元のデータに 0 または 1 の誤り検出用のデータを付加するものがある。たとえば、8 ビットのデータに含まれる 1 の数を数えて、その個数が偶数なら 0 を付加し、奇数なら 1 を付加して全体として 9 ビットのデータとしてデータ伝送を行なう。9 ビットのデータを受け取った方は、含まれる 1 の数を数えて偶数個であれば伝送の際に誤りはなかったと判定し、奇数個であれば伝送の際に 1 ビットが変化した (0 から 1、あるいは 1 から 0) と判定することで、誤り検出が行なえる。もちろん、2 ビット以上の誤りが同時に生じると誤判定になり得る。

ここでは、4 ビットのデータに関して次のように定式化して考える。$x_1x_2x_3x_4$ が伝送する元のデータであり $x_i\,(1 \leqq i \leqq 4)$ は 0 または 1 である。上記の誤り検出のために付加するビットを c とする。$0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0$ という演算を考えることにすると、次式で c が計算できる。

$$c = x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

次に、送られてきたデータ $x_1x_2x_3x_4c$ に誤りが含まれているかどうかを判定する式を考える。次の値 s が 0 であれば誤りはなく、1 であれば誤りがあると考えられる。前述のように 2 ビット以上の誤りについては考えない。

$$s = \boxed{(1)} \oplus \boxed{(2)} \oplus \boxed{(3)} \oplus \boxed{(4)} \oplus \boxed{(5)}$$

(選択肢の番号が、$\boxed{(1)} \leqq \boxed{(2)} \leqq \boxed{(3)} \leqq \boxed{(4)} \leqq \boxed{(5)}$ となるように選ぶこと。)

次に、付加するビットを 3 個にして、1 ビットの誤り訂正ができるような仕組みを考える。次の式で示されるように付加する 3 ビット c_1, c_2, c_3 を決めて、$x_1x_2x_3x_4$ に $c_1c_2c_3$ を付加して 7 ビットのデータ $x_1x_2x_3x_4c_1c_2c_3$ を伝送する。

$$\begin{aligned} c_1 &= \boxed{(6)} \oplus \boxed{(7)} \oplus \boxed{(8)} \\ c_2 &= \boxed{(9)} \oplus \boxed{(10)} \oplus \boxed{(11)} \\ c_3 &= \boxed{(12)} \oplus \boxed{(13)} \oplus \boxed{(14)} \end{aligned}$$

(選択肢の番号が、$\boxed{(6)} < \boxed{(7)} < \boxed{(8)}$、$\boxed{(9)} < \boxed{(10)} < \boxed{(11)}$、$\boxed{(12)} < \boxed{(13)} < \boxed{(14)}$ となるように選ぶこと。)

このように c_1, c_2, c_3 を決めることにすると、1111 は 1111111 として伝送され 1000 は 1000110 として伝送される。元のデータ $x_1x_2x_3x_4$ のすべての可能な組み合わせの数は $\boxed{(15)}\boxed{(16)}$ とおりである。また、

$x_1x_2x_3x_4c_1c_2c_3$ の可能な組み合わせは伝送のときに誤って伝わった場合を含めると (17) (18) (19) とおりである。この (17) (18) (19) とおりのうち伝送のときに誤りがなかったものを除いた (20) (21) (22) とおりは伝送のときに1ビット誤ったものと考え、2ビット以上の誤りは考えないものとする。

x_1 から c_3 までのどれか1ビットが誤って伝送された場合は、誤ったビットの位置によって7種類考えられ、すべてのビットが正しく伝送された場合を合わせて全部で8種類の伝送の結果が考えられる。ここで、送られてきた7ビットからどのビットが誤っているかを示す式があれば誤り訂正が可能となる。8種類の伝送の結果を示すには3ビットあればよいので、送られてきた7ビットから誤り箇所を示す3ビット $(s_1s_2s_3)$ の値を計算できればよい。次の式で s_1, s_2, s_3 を計算すると、$s_1s_2s_3$ の値が0であれば誤りなし、1から7までであればどこかのビットが誤っていることを示す。また、$s_1s_2s_3$ の値に応じて、どのビットが誤っているかが分かる。

$$s_1 = x_1 \oplus x_2 \oplus x_4 \oplus c_1$$
$$s_2 = x_1 \oplus x_3 \oplus x_4 \oplus c_2$$
$$s_3 = x_2 \oplus x_3 \oplus x_4 \oplus c_3$$

たとえば、1101は、1101100として伝送される。1101100から $s_1s_2s_3$ を計算すると000が得られ、誤りが生じなかったことが分かる。次に、4番目のビット x_4 に誤りが生じ1101100を1100100として受け取った場合を考える。1100100から、$s_1s_2s_3$ を計算すると111が得られる。同様に、1000について x_4 が誤って送られたとして、受け取った7ビットから $s_1s_2s_3$ を計算するとやはり111が得られる。上記のように $s_1s_2s_3$ を計算すると、すべての伝送データに対して、x_4 が誤っている場合には常に111が得られようになっていることが分かる。$s_1s_2s_3$ の値とどのビットが誤っているかの対応は次のようになる。

$s_1s_2s_3$	誤ったビット
000	なし
001	(23)
010	(24)
011	(25)
100	(26)
101	(27)
110	(28)
111	(29)

総

【$\boxed{(1)}$〜$\boxed{(14)}$、$\boxed{(23)}$〜$\boxed{(29)}$ の選択肢】

(1) x_1 (2) x_2 (3) x_3 (4) x_4 (5) c

(6) c_1 (7) c_2 (8) c_3 (9) 1 (0) 0

情報Ⅱ

現在広く使われている暦法では、西暦年数を n として、次の規則（「現用ルール」と呼ぶことにする）によって閏年を決めている。

- n が 4 で割り切れれば閏年、そうでなければ平年。
- ただし、n が 4 で割り切れても、100 で割り切れる場合は平年。
- ただし、n が 100 で割り切れても、400 で割り切れる場合は閏年。

(ア) 次の 3 つのプログラムは、いずれも現用ルールに従って西暦 1 年から 400 年までが平年か閏年かを順に表示するものである。空欄 (30) ～ (35) に当てはまるもっとも適切な選択肢を下から選び、その番号を解答欄にマークしなさい。

```
nを1から400まで1ずつ増やしながら繰り返す :
│ もし (30) ならば :
│ │ 表示する("平年")
│ そうでなければ :
│ │ もし (31) ならば :
│ │ │ 表示する("閏年")
│ │ そうでなければ :
│ │ │ もしn % 400 != 0ならば :
│ │ │ │ 表示する("平年")
│ │ │ そうでなければ :
└ └ └ └ 表示する("閏年")
```

```
nを1から400まで1ずつ増やしながら繰り返す :
│ x = "平年"
│ もし (32) ならば :
│ └ x = "閏年"
│ もしn % 100 == 0ならば :
│ └ x = "平年"
│ もし (33) ならば :
│ └ x = "閏年"
└ 表示する(x)
```

```
n を 1 から 400 まで 1 ずつ増やしながら繰り返す ：
│ もし ( n % 4 == 0 and [(34)] ) or [(35)] ならば ：
│ │ 表示する ("閏年")
│ そうでなければ ：
└ └ 表示する ("平年")
```

【[(30)] ～ [(35)] の選択肢】

(1) n % 4 == 0　(2) n % 100 == 0　(3) n % 400 == 0

(4) n % 4 != 0　(5) n % 100 != 0　(6) n % 400 != 0

(イ) 空欄 [(36)][(37)][(38)] ～ [(50)][(51)][(52)] に当てはまるもっとも適切な数字を解答欄にマークしなさい。

閏年が必要なのは、1 年が 1 日の整数倍ではなく、暦の 1 年を常に 365 日とすると、暦の日付と太陽の位置がずれてしまうからである。次のプログラムは、1 年を 365.2422 日として西暦 1 年から積算し、暦とのずれがある定数 a を超えたら、その年を閏年とするものである。ただし、a の値はプログラム実行前に決め、実行中は変化しないものとする。また、演算は十分な精度の 10 進固定小数点で行い、10 進 2 進変換や浮動小数点演算による誤差はないものとする。

```
d = 0
n を 1 から 400 まで 1 ずつ増やしながら繰り返す ：
│ d = d + 365.2422
│ もし d > 365 + a ならば ：
│ │ 表示する ("閏年")
│ │ d = d - 366
│ そうでなければ ：
│ │ 表示する ("平年")
└ └ d = d - 365
```

例えば、a の値が 0.5 の場合、西暦 1 年から 4 年については「平年、平年、閏年、平年」と表示される。

- a の値が 1 の場合、西暦 1 年から 400 年の間に「閏年」は [(36)][(37)][(38)] 回表示される。
- a の値が 2 の場合、西暦 1 年から 400 年の間に「閏年」は [(39)][(40)][(41)] 回表示される。
- 西暦 1 年から 20 年について現用ルールと一致するのは、a の値が 0.[(42)][(43)][(44)][(45)] ≦ a < 0.[(46)][(47)][(48)][(49)] の場合である。
- 西暦 1 年からなるべく長く現用ルールと一致するように a の値を決めると、西暦 [(50)][(51)][(52)] 年まで一致させることができる。

情報Ⅲ

マス目で構成された迷路状のコースを走行する移動ロボットのコンピュータのプログラムや動作について考える。このロボットは、センサー、コンピュータ、駆動輪から構成されている。センサーは、ロボットがいるマスにおける左側、前方、右側の壁の有無と、そのマスがゴールのマスであるか否かを検知する。コンピュータは、センサーからの情報を取得し、ロボットをその場で旋回させるか、前進させるかを決定し、駆動輪に指令する。駆動輪は、コンピュータからの指令に従いロボットを旋回あるいは前進させる。

センサー情報の取得、駆動輪による移動に関しては、以下の関数が用意されているものとする。

壁センサー情報取得 () ロボットがいるマスの壁の有無（左側、前方、右側）の情報を取得する。返り値は整数で、壁の有無と返り値の関係を表１に示す。

表１

左側の壁	前方の壁	右側の壁	返り値
無し	無し	無し	0
無し	無し	有り	1
無し	有り	無し	2
無し	有り	有り	3
有り	無し	無し	4
有り	無し	有り	5
有り	有り	無し	6
有り	有り	有り	7

ゴールセンサー情報取得 () ロボットがいるマスがゴールであるか否かの情報を取得する。ロボットがいるマスがゴールである場合の返り値は 1、ゴールでない場合の返り値は 0 となる。

左旋回 (n)

- n が 0 の場合、その場で左方向に 90 度、ロボットの向きを変えるよう指令し、駆動輪の動作が完了するまで待機する。返り値は 1 となる。
- n が 0 以外の場合、何もせず、返り値は 0 となる。

右旋回 (n)

- n が 0 の場合、その場で右方向に 90 度、ロボットの向きを変えるよう指令し、駆動輪の動作が完了するまで待機する。返り値は 1 となる。
- n が 0 以外の場合、何もせず、返り値は 0 となる。

前進 (n)

- n が 0 かつ前方に壁が無い場合、ロボットを 1 マス前進させるよう指令し、駆動輪の動作が完

了するまで待機する。返り値は 1 となる。

- n が 0 かつ前方に壁がある場合、何もせず、返り値は 0 となる。
- n が 0 以外の場合、何もせず、返り値は 0 となる。

図 1 は、2×2 のマス目で構成されるコースとロボットを示したものである。迷路を正方形のマス目で示し、マス目の辺が実線の部分は壁があることを示し、点線の部分は壁が無いことを示している。「G」のマークがついているマスがゴールのマスであること示している。グレーの長方形がロボットの位置、内部の矢印が向きを示している。矢印の方向がロボットの前方の方向である。

ロボットが図 1 の A に示す位置の場合、**壁センサー情報取得** () の返り値は 5 となる。ここで、**前進** (0) を実行すると 1 マス前進して、図 1 の B に示す位置になる。この位置で**壁センサー情報取得** () の返り値は 3 となる。ここで**左旋回** (0) を実行すると、その場で向きが変わり図 1 の C に示す向きとなる。この位置で**壁センサー情報取得** () の返り値は 1 となる。ここで **前進** (0) を実行すると、図 1 の D に示す位置となりロボットはゴールに到着する。**ゴールセンサー情報取得** () の返り値は、図 1 の A、B、C に示す位置では 0、図 1 の D に示す位置では 1 となる。

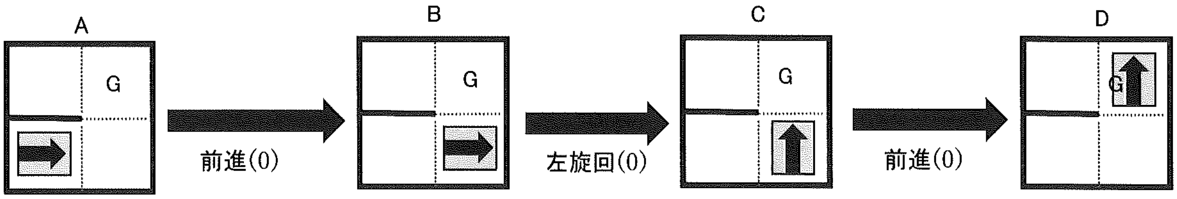

図 1

(ア) 空欄 (55) ～ (58) に入る最も適切なものを下の選択肢から選び、その番号を解答欄にマークしなさい。また、空欄 (53)、(54) には、当てはまるもっとも適切な数字を解答欄にマークしなさい。

図 2 のようなコースをロボットに走行させゴールに到着させるためのプログラムを作成する。図 2 の X, Y はマスの位置を示すための座標系である。ロボットは、X=0, Y=0 のマスから矢印の方向を向いてスタートする。ゴールは、X=0, Y=1 のマスである。

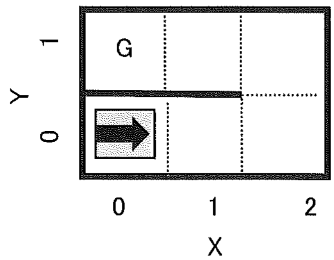

図 2

ロボットのプログラムとして以下のプログラム P1 あるいは P2 を使用した場合、ロボットはゴールに到達し、「ゴール到着」と表示され、繰り返し処理の回数 (変数 step の値) が表示される。その回数

は、P1 を使用した場合は (53)、P2 を使用した場合は (54) である。

【プログラム P1】

```
step = 0
ゴールセンサー情報取得 () == 0 の間繰り返す :
│ s = 壁センサー情報取得 ()
│ もし s < 4 ならば :
│ └ (55)
│ (56)
└ step = step + 1
表示する ("ゴール到着 繰り返し処理の回数は", step)
```

【プログラム P2】

```
step = 0
ゴールセンサー情報取得 () == 0 の間繰り返す :
│ t = 前進 (0)
│ t == 1 の間繰り返す :
│ └ t = (57)
│ (58)
└ step = step + 1
表示する ("ゴール到着 繰り返し処理の回数は", step)
```

【(55) ～ (58) の選択肢】

(1) **前進** (0)	(2) **前進** (1)
(3) **左旋回** (0)	(4) **左旋回** (1)
(5) **右旋回** (0)	(6) **右旋回** (1)
(7) **壁センサー情報取得** ()	(8) **ゴールセンサー情報取得** ()

(イ) 次の文章の空欄 (59) ～ (62) に当てはまるもっとも適切な数字を解答欄にマークしなさい。

図 3 のような迷路状のコースをロボットに走行させ、ゴールに到達させることを試みる。ロボットは、X=0, Y=0 のマスから矢印の方向を向いてスタートする。ゴールは、X=0, Y=1 のマスである。

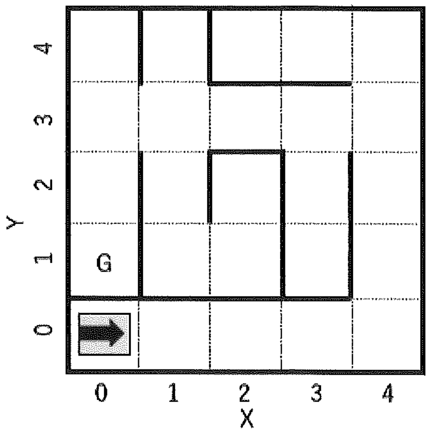

図 3

図 3 のコースをプログラム P1 で走行させた場合、あるマスで身動きが取れなくなる。身動きが取れなくなるマスの X 座標は (59)、Y 座標は (60) である。

また、図 3 のコースをプログラム P2 で走行させた場合、ゴールに到達せず、ある区間を往復し続けることになる。この区間のマスの中で、X 座標と Y 座標の合計値が最も小さいマスの X 座標は (61)、Y 座標は (62) である。

(ウ) 次の文章の空欄 (63)(64) に当てはまるもっとも適切な数字を解答欄にマークしなさい。

図 3 のコースのゴールに到達させるためのプログラムとして、プログラム P3 を作成したら、無事にゴールに到達できることを確認した。その時に表示される繰り返し処理の回数は (63)(64) である。

【プログラム P3】

```
step = 0
ゴールセンサー情報取得 () == 0 の間繰り返す :
│ s = 壁センサー情報取得 ()
│ a = 右旋回 (s % 2)
│ もし a == 0 ならば :
│ │ b = 前進 (0)
│ │ もし b == 0 ならば :
│ │ │ c = 左旋回 (s ÷ 4)
│ │ │ もし c == 0 ならば :
│ │ │ │ 左旋回 (0)
│ └ └ └ 左旋回 (0)
│ 前進 (0)
└ step = step + 1
表示する ("ゴール到着 繰り返し処理の回数は", step)
```

（計算用紙）

数学の解答は解答用紙の解答欄 (76)〜(135) にマークしてください.

数学Ⅳ

立方体の各面に色を塗る方法を考える．ただし，立方体を回転して一致する塗り方は同じとみなす．各面には 1 色しか塗らないものとする．

(1) 赤，青，緑，黄，白，黒の 6 色をすべて使って塗る方法は | (76) | (77) | (78) | 通りある．

(2) 赤，青，緑，黄，紫，茶，白，黒の 8 色から異なる 6 色を選び，その 6 色をすべて使って塗る方法は | (79) | (80) | (81) | 通りある．

(3) 白と黒の 2 色をすべて使って塗る方法は | (82) | (83) | (84) | 通りある．

(4) 赤，青，黄，白，黒の 5 色をすべて使い，隣り合う面の色が異なるように塗る方法は | (85) | (86) | (87) | 通りある．

(5) 6 面のうち 2 面を赤で塗り，青，黄，白，黒の 4 色をすべて使って残り 4 面を塗る方法は | (88) | (89) | (90) | 通りある．

（計算用紙）

数学V

実数 a, b, c に対して，関数 $f(x)$ と $g(x)$ は次の等式を満たしている．

$$f(x) - \int_0^x (3t^2 + 2t)\, g'(t)\, dt = 3x^3 + ax^2 + bx + 1$$

$$g(x) + 6\int_1^x t\, f'(t)\, dt = 8x^3 - 21x^2 + cx + 20$$

このとき

$$f(x) = \boxed{(91)\,\vert\,(92)}\, x^2 + \boxed{(93)\,\vert\,(94)}\, x + \boxed{(95)\,\vert\,(96)}$$

$$g(x) = \boxed{(97)\,\vert\,(98)}\, x + \boxed{(99)\,\vert\,(100)}$$

であり，$a = \boxed{(101)\,\vert\,(102)}$，$b = \boxed{(103)\,\vert\,(104)}$，$c = \boxed{(105)\,\vert\,(106)}$ である．

また，xy 平面上で，曲線 $y = f(x)$ と直線 $y = g(x)$ によって囲まれた図形の面積は $\dfrac{\boxed{(107)\,\vert\,(108)}}{\boxed{(109)\,\vert\,(110)}}$ である．

（計算用紙）

総

数学VI

日本のジャンケンは，石 (グー)，はさみ (チョキ)，紙 (パー) の 3 種類のどれかを出して勝ち負けを決めるゲームだが，フランスのジャンケンは，石 (ピエール)，井戸 (ピュイ)，木の葉 (フェイユ)，はさみ (シゾー) の 4 種類のどれかの手を出して勝ち負けを決める．

石とはさみでは，はさみを切れなくする石が勝ち，はさみが負ける．はさみと木の葉では，木の葉を切るはさみが勝ち，木の葉が負ける．石とはさみは井戸に落ちるので，井戸は石とはさみに勝ち，石とはさみは井戸に負ける．木の葉は，石を包み井戸をふさぐことができるので石と井戸に勝ち，石と井戸は木の葉に負ける．また，同じ手の場合にはあいことなり引き分けとなる．

いま，A 君と B 君がフランスのジャンケンで勝ち負けを決めようとしているが，フランスのジャンケンの場合には出す手の戦略をいくつか考えることができる．

(戦略 1) 石，井戸，木の葉，はさみを $\frac{1}{4}$ ずつの確率で出す．

(戦略 2) 井戸と木の葉の両方に負ける石を出さずに，井戸，木の葉，はさみを $\frac{1}{3}$ ずつの確率で出す．

(戦略 3) 石と井戸の両方に負けるはさみを出さずに，石，井戸，木の葉を $\frac{1}{3}$ ずつの確率で出す．

(戦略 4) 2 つの手に負ける石とはさみを出さずに，井戸と木の葉を $\frac{1}{2}$ ずつの確率で出す．

(1) A 君が戦略 2 で手を出し，B 君が戦略 1 で手を出すとき，1 回のジャンケンで A 君が勝つ確率は $\dfrac{\boxed{(111)}\boxed{(112)}}{\boxed{(113)}\boxed{(114)}}$ で，B 君が勝つ確率は $\dfrac{\boxed{(115)}\boxed{(116)}}{\boxed{(117)}\boxed{(118)}}$ である．

(2) A 君が戦略 4 で手を出し，B 君が戦略 3 で手を出すとき，1 回のジャンケンで A 君が勝つ確率は $\dfrac{\boxed{(119)}\boxed{(120)}}{\boxed{(121)}\boxed{(122)}}$ で，B 君が勝つ確率は $\dfrac{\boxed{(123)}\boxed{(124)}}{\boxed{(125)}\boxed{(126)}}$ である．

(3) B 君が戦略 1 で手を出すとき，A 君は 4 つの戦略のうち戦略 $\boxed{(127)}$ を選べば，1 回のジャンケンで A 君の勝つ確率が $\dfrac{\boxed{(128)}\boxed{(129)}}{\boxed{(130)}\boxed{(131)}}$ で，B 君が勝つ確率が $\dfrac{\boxed{(132)}\boxed{(133)}}{\boxed{(134)}\boxed{(135)}}$ となり，A 君の勝つ確率と B 君の勝つ確率の比 (引き分けは考えずに A 君の勝つ確率を B 君の勝つ確率で割ったもの) を最も大きくすることができる．

（計算用紙）

（計算用紙）