

解答用紙の選択科目名に「情報」と記入し、選択科目マーク欄の「情報」をマークしてから解答してください。情報の解答は解答用紙の解答欄(1)～(100)にマークしてください。

情報Ⅰ

以下、法制度に関しては、日本のものについて考えるものとする。

次の文章を読み、設問に回答しなさい。

著作権法上「著作者」とは「著作物を創作する者」と定義されている。また、1つの著作物の作成に複数の者が関与している場合には、著作物の作成に創作的に寄与した者が著作者であると考えられる（著作権法第2条第1項第12号（共同著作物）参照）。

コンピュータ創作物の作成過程に関与する者としては、1) コンピュータ・システムの使用者、2) コンピュータ・システムにおいて実行されるプログラムの作成者、3) データ又はデータベースなどの形でコンピュータ・システムに入力される素材の作成者が考えられる。（中略）

(1) コンピュータ・システムの使用者

コンピュータ創作物に著作物性が認められる場合、その著作者は具体的な結果物の作成に創作的に寄与した者と考えられるが、通常の場合、それは、コンピュータ・システムの使用者であると考えられる。^①

ただし、使用者が単なる操作者であるにとどまり、何ら創作的寄与が認められない場合には、当該使用者は著作者とはなり得ない。どのような場合に使用者が創作的寄与を行ったと評価でき、又は単なる操作者にとどまるかについては、個々の事例に応じて判断せざるを得ないが、一般に使用者の行為には入力段階のみならず、その後の段階においても^①形式などにより各種の処理を行い、最終的に一定の出力がなされたものを選択して作品として固定するという段階があり、これらの一連の過程を総合的に評価する必要がある。

(2) プログラムの作成者

プログラムの作成者は、プログラムがコンピュータ・システムとともに使用者により創作行為のための^②として用いられるものであると考えられるため、一般的には、コンピュータ創作物の著作者とはなり得ないと考えられる（例えば、OSや汎用的プログラムの作成者）。

ただし、プログラムの作成行為と使用者の創作行為に共同性が認められるとするならば、プログラムの作成者がコンピュータ・システムの使用者と共に共同著作者となる場合もあり得ると考えられる（例

例えば、使用者とプログラマーが特定の創作物を共同して創作する意図の下に共同作業計画を策定し、それを踏まえてプログラマーが特定の創作物作成の用に供するためのプログラムを作成する場合)。

また、プログラムの作成者が自ら特定の創作物の作成を意図して、そのために作成されたものであると客観的に認識できる程度の特定性があるプログラムを作成し、使用者は単なる (3) 者にとどまる場合には、当該プログラムの作成者が単独でコンピュータ創作物の著作権となることもあり得ると考えられる。

(3) 素材の作成者

データ又はデータベースなどの形でコンピュータ・システムに入力される素材の作成者は、素材自体が (4) 物であり、コンピュータ創作物がその (5) 著作物に当たる場合には、原著作物の著作権たる地位を有するが、一般的には、コンピュータ創作物自体の著作権とはなり得ないと考えられる。

ただし、プログラムの場合と同様、素材の作成行為と使用者の創作行為に共同性が認められるとするならば、素材の作成者がコンピュータ・システムの利用者と共に共同著作権となる場合もあり得ると考えられる。

また、素材の作成者が、プログラムの作成者と共同して特定の創作物の作成を意図して、そのための特定のプログラム及び素材を作成していると認められ、使用者は単なる (3) 者であるに留まる場合には、素材の作成者とプログラムの作成者が共同著作権となる場合もあり得ると考えられる。

なお、素材の作成者が単独でコンピュータ創作物の著作権となることはほとんどあり得ないと考えられる。②（後略）

（出典：文化庁「著作権審議会第9小委員会（コンピュータ創作物関係）報告書」（1993年）を一部改変）

(ア) 空欄 (1) ～ (5) に入るもっとも適した語を選択肢から選び、その番号を解答欄にマークしなさい。

- (1) 創作 (2) 有体 (3) 操作 (4) 道具 (5) アイデア
(6) バイナリ (7) 対話 (8) 二次的 (9) 著作 (0) 共同

(イ) 下線部①の理由として、もっとも適切なものを選択肢から選び、その番号を解答欄 (6) にマークしなさい。

- (1) コンピュータ・システムの利用者は、物理的な操作を実施しているから。
(2) コンピュータ・システムの利用者は、ハードウェアの所有権を有しているから。

- (3) コンピュータ・システムの利用者は、対価を支払ってプログラムの使用権を購入しているから。
- (4) コンピュータ・システムの利用者は、システムの使用法や結果物を選択して作品としての固定を決定しているから。
- (5) コンピュータ・システムの利用者は、プログラムの作成者と共同して作業を担当しているから。

(ウ) 下線部②の理由として、もっとも適切なものを選択肢から選び、その番号を解答欄 (7) にマークしなさい。

- (1) 素材の作成者は、コンピュータ創作物の生成に際して付加される属性の内容に寄与していないから。
- (2) 素材の作成者は、コンピュータ・システムの所有権を有していないから。
- (3) 素材の作成者は、コンピュータ・システムの利用者から素材の対価を受け取っているから。
- (4) 素材の作成者は、コンピュータ創作物の生成を認識していない場合があるから。
- (5) 素材の作成者は、プログラムの作成者と共同して作業していない場合があるから。

情報Ⅱ

次の文章の空欄

(8)	(9)
-----	-----

 ～

(18)	(19)
------	------

 に入るもっとも適したものを選択肢から選び、解答欄にマークしなさい。また、空欄

(20)	(21)	(22)
------	------	------

 ～

(23)	(24)	(25)
------	------	------

 に入るもっとも適した数字を解答欄にマークしなさい。

一定時間内に切符売り場や ATM などの窓口にやって来る人の人数とそのモデル化について考える。窓口にはランダムな間隔で人が来るものとし、個々の到着は独立していて、依存関係はないものとする。ここで、単位時間あたりにどれだけの人が来るかの率を到着率と呼び、 λ で表す。たとえば、平均して 100 秒に 1 人の到着率で人が来る場合、 λ は 0.01 人毎秒になり、1000 秒間に来る人の数の期待値は 10 人になる。ここで、 λ に対して十分に小さい時間、たとえば 1 秒を考えると、その間に人が来る確率はおよそ 100 分の 1 と考えられる。しかしながら、一定間隔で規則的に来るわけではないので、この場合、ある 100 秒の間に人が来る確率は 1 とはならないことに注意しなければならない。また、より長い時間を考えた場合、1000 秒の間にやってくる人数は 10 人とはかぎらず、ある 1000 秒間では 9 人だったり、別の 1000 秒間では 11 人だったりする。つまり、ある一定時間内に来る人の数の期待値は λ に時間を乗じたものになるが、ある一定時間内に来る人の数は、期待値どおりにはならない。このことは、コインを 10 回投げたときに表の出る回数の期待値が 5 回と考えられるが、10 回投げたときにいつも表が 5 回出るのではないことを考えると理解しやすい。

ここでは、このようなモデルにおける一定時間内に来る人数の分布を考える。このとき、前述のように、十分に短い時間に 1 人が到着する確率は到着率に比例し、2 人以上がこの短い時間内に来る確率は無視できるほど小さいと考える。そこで、十分短い時間 Δt の間に窓口の人が来る確率は、 $\lambda \Delta t$ になるとする。

ここで、時間 t の間に k 人が窓口に来る確率 $p_k(t)$ を考える。 t を n 個に分割し、 $\Delta t = t/n$ とする。そして、この n 個の短い時間 Δt のうち、どの時間に人が来て、どの時間には人が来ないかの確率を考える。 n 個の中の特定の k 個で窓口の人が来て、残りの $(n - k)$ 個で窓口の人が来ない確率は、次のようになる。

$$\left(\begin{array}{|c|c|} \hline (8) & (9) \\ \hline \end{array} \right)^k \left(\begin{array}{|c|c|} \hline (10) & (11) \\ \hline \end{array} \right)^{n-k}$$

ここで、 k 個の選び方は、 ${}_nC_k$ 通りあるので、

$$p_k(t) = {}_nC_k \left(\begin{array}{|c|c|} \hline (8) & (9) \\ \hline \end{array} \right)^k \left(\begin{array}{|c|c|} \hline (10) & (11) \\ \hline \end{array} \right)^{n-k}$$

が得られる。これを变形すると、

$$p_k(t) = \left(1 - \frac{\lambda t}{n}\right)^n \frac{(\boxed{(12)} \boxed{(13)})^k}{k!} \left(1 - \frac{\lambda t}{n}\right)^{\boxed{(14)} \boxed{(15)}} \frac{n!}{n^k (\boxed{(16)} \boxed{(17)})!}$$

となり、ここで十分大きな n を考えると次の式が得られることが知られている。

$$p_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

ここで、底 e は無理数になるが、以下の説明で必要な場合には近似値として $e = 2.718$, $e^{-1} = 0.3679$, $e^{-2} = 0.1353$ を用いて計算する。

また、 $p_k(t)$ は確率なので、全事象の確率の和を計算すると 1 になり、次式が成り立つ。

$$\sum_{k=0}^{\infty} p_k(t) = 1$$

同様に、各事象の確率とその事象における到着人数の積の和は、時間 t の間に来る人の数の期待値になっている。この値は、最初に説明した一定時間内に来る人の数の期待値に等しく、次式が得られる。

$$\sum_{k=0}^{\infty} p_k(t) k = \boxed{(18)} \boxed{(19)}$$

次にこのモデルを用いた計算例を示す。到着率 λ を 10 分あたり 2 人とする。この場合、ある 10 分間に 1 人も来ない確率を小数第 4 位を四捨五入して小数第 3 位まで求めると、0. $\boxed{(20)} \boxed{(21)} \boxed{(22)}$ になる。また 1 人だけ来る確率を小数第 4 位を四捨五入して小数第 3 位まで求めると、0. $\boxed{(23)} \boxed{(24)} \boxed{(25)}$ になる。

【 $\boxed{(8)} \boxed{(9)} \sim \boxed{(18)} \boxed{(19)}$ の選択肢】

- | | | | | |
|----------------------|-------------------------|-----------------------------|-----------------------------|------------------|
| (11) n | (12) $-n$ | (13) $n - 1$ | (14) $n + 1$ | (15) k |
| (16) $-k$ | (17) $k + 1$ | (18) $k - 1$ | (19) 0 | (20) 1 |
| (21) $n - k$ | (22) $k - n$ | (23) $n!$ | (24) $(n - 1)!$ | (25) $k!$ |
| (26) $(k - 1)!$ | (27) $\lambda \Delta t$ | (28) $\lambda \Delta t - 1$ | (29) $1 - \lambda \Delta t$ | (30) λt |
| (31) $1 - \lambda t$ | (32) $\lambda t - 1$ | (33) $\frac{k}{n}$ | (34) $\frac{n}{k}$ | |

情報Ⅲ

空欄

(26)	(27)	(28)
------	------	------

 ～

(49)

 に入るもっとも適した数字を解答欄にマークしなさい。

(ア) 16 進数 CD と 2 進数 0111 を加えた数は 8 進数で表現すると

(26)	(27)	(28)
------	------	------

 となる。また、8 進数 57 と 16 進数 3B をそれぞれ 2 進数として表現したものについて、各桁を真偽値として論理積を計算し、その結果を各桁とする 2 進数を 10 進数として表現すると

(29)	(30)	(31)
------	------	------

 となる。ただし、0 は偽、1 は真を表しているものとする。

(イ) 生徒が 500 人いる学校において、生徒番号と氏名、それぞれの科目の成績が格納されているデータセットがあり、それに対して生徒番号から成績を検索するシステムを考える。生徒番号は生徒 1 人に対して 1 つ与えられ、重複はないものとする。検索対象の生徒番号が与えられると、順にデータセットの中の生徒番号と等しいかどうか比較し、等しいものが見つければ直ちにその成績を出力する。等しいものが見つからなかった場合は、「その生徒番号は存在しない」と出力する。検索対象の生徒番号がランダムに与えられ、与えられる生徒番号は 0.1 の確率でデータセットに存在しない場合において、生徒番号の比較が行われる回数の平均は

(32)	(33)	(34)
------	------	------

 .

(35)	(36)
------	------

 である。ただし、小数第 3 位以下が存在する場合は小数第 3 位を四捨五入せよ。

(ウ) 30 人のクラスにおいて、クラス内に同じ誕生日の人が少なくとも 2 人以上いる場合の確率は以下のように表現される。ただし、1 年は 365 日とし、誕生日の分布は一様であるものとする。

$$1 - \frac{{}_aP_{29}}{b^c}$$

$$a = \begin{array}{|c|c|c|} \hline (37) & (38) & (39) \\ \hline \end{array}, b = \begin{array}{|c|c|c|} \hline (40) & (41) & (42) \\ \hline \end{array}, c = \begin{array}{|c|c|c|} \hline (43) & (44) & (45) \\ \hline \end{array}$$

(エ) A, B, C のラベルがついた 3 つのふたがされた箱があり、その中から賞品が入っている箱を当てるゲームを考える。3 つの箱のうち 1 つの箱には賞品が入っており、残りの 2 つは空であることがわかっている。

最初に参加者は A の箱を選択すると宣言した。この時 A の箱も他の箱も開けられていない状態で、正解を知っているスタッフが C の箱のふたを開け、中が空であることを示した。この時点で参加者に

は選択した箱を変更する権利が与えられる。箱 A および B に賞品が入っている確率はそれぞれいくらか。なお最初の段階で A, B, C の箱には等しい確率で賞品が入れられるものとする。

- 箱 A に賞品が入っている確率 $\frac{\boxed{(46)}}{\boxed{(47)}}$
- 箱 B に賞品が入っている確率 $\frac{\boxed{(48)}}{\boxed{(49)}}$

情報Ⅳ

次の文章の空欄 (70) (71) (72) にあてはまる数字を解答欄にマークしなさい。また、(50) ～ (68) (69) にはもっとも適したものを選択肢から選び、解答欄にマークしなさい。ただし、 $A + B$ は A と B の論理和 (OR) を表し、 $A \cdot B$ は A と B の論理積 (AND) を表す。また、 \overline{A} は A の否定 (NOT) を表す。

コンピュータを構成する基本的な装置のひとつに、算術論理演算装置 (Arithmetic Logic Unit、以下 ALU と表記) がある。ここでは、図 1 に示す 4 種類の演算を実行する簡単な ALU を設計しよう。

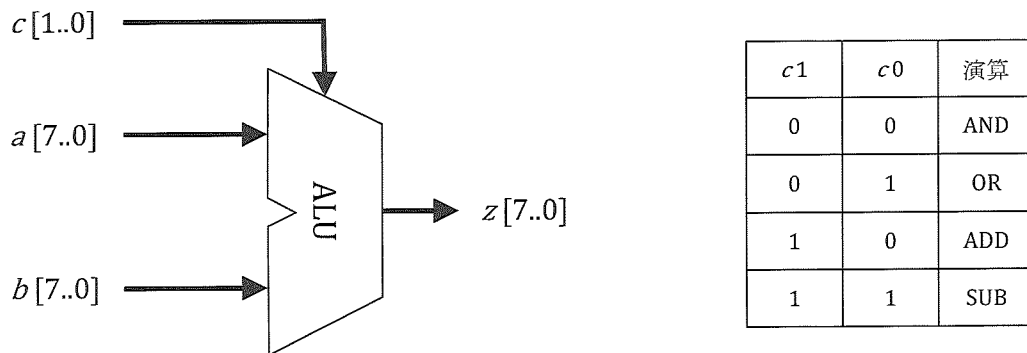


図 1

4 種類の演算とは、論理積 (AND)、論理和 (OR)、算術演算として整数の加減算 (ADD、SUB) である。ALU への入力は、2 組の 8 ビットのデータ $a[7..0]$ 、 $b[7..0]$ および 2 ビットの制御信号 $c[1..0]$ であり、出力は 8 ビットのデータ $z[7..0]$ である。ここで、 $c[1..0]$ は 2 個の信号 $c1$, $c0$ をまとめて表記したものである。同様に $a[7..0]$ は 8 個の信号をまとめて表記しており、算術演算 (ADD、SUB) を行うときは、信号 (0 または 1) を並べたビット列 $a7\ a6\ a5\ a4\ a3\ a2\ a1\ a0$ を 2 進法表現による整数と考える。 $b[7..0]$ 、 $z[7..0]$ についても同じである。

4 種類の演算のうち、どの演算の結果が出力されるかは、 $c[1..0]$ により決定される。 $c[1..0]$ と実行される演算の対応を図 1 右の表に示す。

(ア) この ALU の回路構造は図 2 のようになる。AND、OR は 8 ビットのデータ同士のビットごとの論理積および論理和を実行する論理回路モジュールであり、ADD_SUB は 8 ビットの整数同士の算術加減算回路モジュールである。1 ビットの制御信号 ($subc$) が 1 のとき減算 $a[7..0] - b[7..0]$ を実行し、0 の場合は加算 $a[7..0] + b[7..0]$ が実行されるものとする。図 2 中 ア は (50) である。

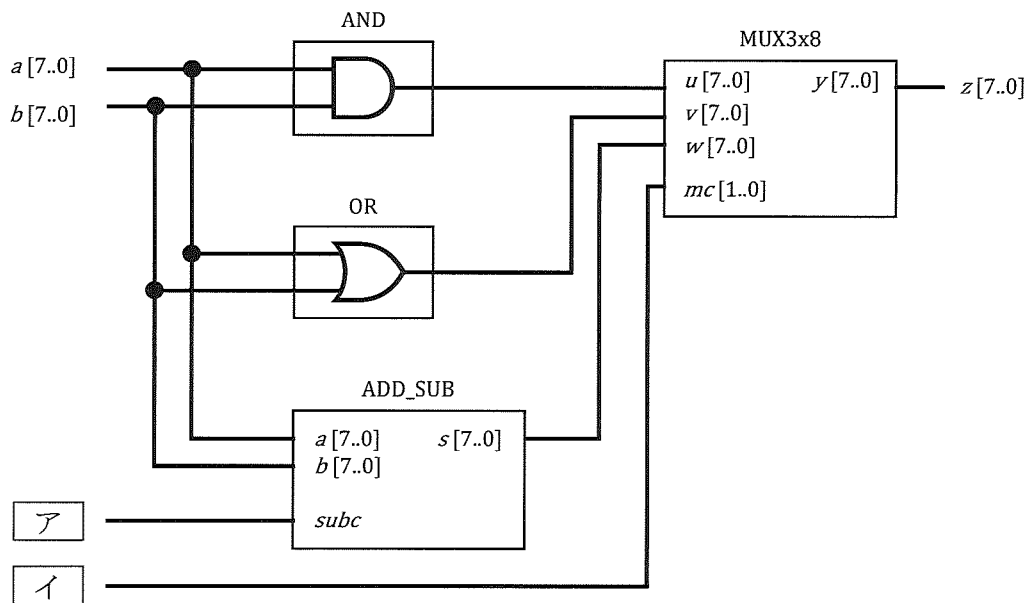


図 2

【(50) の選択肢】

(1) $c0$ (2) $c1$ (3) $c[1..0]$

(イ) AND、OR、ADD_SUB モジュールの出力である 8 ビット幅データを入力とし ALU としての出力を選択する回路モジュールはマルチプレクサ（データセレクタ）と呼ばれ、図 2 中では MUX3x8 と記載されている。MUX3x8 は、それぞれ 8 ビットの 3 つの入力から 1 つを選択し出力する 3-to-1 マルチプレクサであり、制御信号 $mc[1..0]$ により、動作表（表 1）に示すように出力が制御される。図 2 中「イ」は (51) である。

表 1

$mc1$	$mc0$	y
0	0	u
0	1	v
1	0	w
1	1	w

図 3 に示す、2 つの入力の一方を選択し出力する 1 ビットの 2-to-1 マルチプレクサを考える。制御信号 MC が 0 のときに $A0$ が選択され、 MC が 1 のときには $A1$ が選択されるものとする、動作をま

とめた真理値表は表 2 のようになる。ただし、 ϕ は 0 と 1 のどちらでもよい (don't care) を意味する。

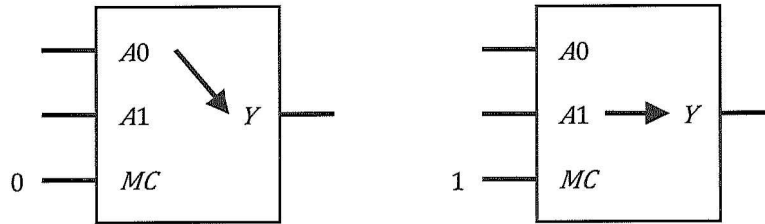


図 3

表 2

MC	A0	A1	Y
0	(52)	(53)	0
0	(54)	(55)	1
1	(56)	(57)	0
1	(58)	(59)	1

この真理値表より、1 ビット 2-to-1 マルチプレクサの論理式は次のように導かれる。((60) (61) 、

(62) (63) は順不同)

$$Y = \begin{matrix} (60) & (61) \end{matrix} + \begin{matrix} (62) & (63) \end{matrix} \quad (1)$$

この 1 ビット 2-to-1 マルチプレクサを入力ビットごとに 8 個準備し、同じ制御信号で動作させることで 8 ビット 2-to-1 マルチプレクサを構成することができる。

同様の設計方針から、上記の ALU で必要となる 1 ビット 3-to-1 マルチプレクサの論理式は、入力を $A0$ 、 $A1$ 、 $A2$ 、出力を Y 、制御信号を $MC0$ 、 $MC1$ とすると、表 1 の $mc0$ 、 $mc1$ 、 u 、 v 、 w 、 y を $MC0$ 、 $MC1$ 、 $A0$ 、 $A1$ 、 $A2$ 、 Y と読み替えることにより

$$Y = \begin{matrix} (64) & (65) \end{matrix} + \begin{matrix} (66) & (67) \end{matrix} + \begin{matrix} (68) & (69) \end{matrix} \quad (2)$$

となり ((64) (65) 、 (66) (67) 、 (68) (69) は順不同)、図 1 の ALU で必要となる 8 ビット 3-to-1 マルチプレクサにおける AND、OR、NOT 回路の合計は (70) (71) (72) 個となる。ただし、回路内で使用する AND および OR に相当する論理回路は 2 入力 1 出力であるとする。

【(51) の選択肢】

(1) $c0$ (2) $c1$ (3) $c[1..0]$

【 $\boxed{(52)} \sim \boxed{(59)}$ の選択肢】

(0) 0 (1) 1 (2) ϕ

【 $\boxed{(60)} \boxed{(61)} \sim \boxed{(62)} \boxed{(63)}$ の選択肢】

(11) MC (12) $A0$ (13) $A1$ (14) \overline{MC}

(15) $\overline{A0}$ (16) $\overline{A1}$ (17) $MC \cdot A0$ (18) $MC \cdot A1$

(19) $A0 \cdot A1$ (20) $\overline{MC} \cdot A0$ (21) $\overline{MC} \cdot A1$ (22) $MC \cdot \overline{A0}$

(23) $MC \cdot \overline{A1}$ (24) $\overline{MC} \cdot \overline{A0}$ (25) $\overline{MC} \cdot \overline{A1}$ (26) $\overline{MC} \cdot \overline{A0} \cdot \overline{A1}$

【 $\boxed{(64)} \boxed{(65)} \sim \boxed{(68)} \boxed{(69)}$ の選択肢】

(11) $MC0 \cdot A0$ (12) $MC0 \cdot A1$ (13) $MC0 \cdot A2$

(14) $MC1 \cdot A0$ (15) $MC1 \cdot A1$ (16) $MC1 \cdot A2$

(17) $\overline{MC0} \cdot A0$ (18) $\overline{MC0} \cdot A1$ (19) $\overline{MC0} \cdot A2$

(20) $\overline{MC1} \cdot A0$ (21) $\overline{MC1} \cdot A1$ (22) $\overline{MC1} \cdot A2$

(23) $MC0 \cdot \overline{A0}$ (24) $MC0 \cdot \overline{A1}$ (25) $MC0 \cdot \overline{A2}$

(26) $MC1 \cdot \overline{A0}$ (27) $MC1 \cdot \overline{A1}$ (28) $MC1 \cdot \overline{A2}$

(29) $MC0 \cdot MC1 \cdot A0$ (30) $MC0 \cdot MC1 \cdot A1$ (31) $MC0 \cdot MC1 \cdot A2$

(32) $\overline{MC0} \cdot MC1 \cdot A0$ (33) $\overline{MC0} \cdot MC1 \cdot A1$ (34) $\overline{MC0} \cdot MC1 \cdot A2$

(35) $MC0 \cdot \overline{MC1} \cdot A0$ (36) $MC0 \cdot \overline{MC1} \cdot A1$ (37) $MC0 \cdot \overline{MC1} \cdot A2$

(38) $\overline{MC0} \cdot \overline{MC1} \cdot A0$ (39) $\overline{MC0} \cdot \overline{MC1} \cdot A1$ (40) $\overline{MC0} \cdot \overline{MC1} \cdot A2$

情報V

フィボナッチ数列とは、 $a_1 = 1$, $a_2 = 1$, $a_n = a_{n-1} + a_{n-2}$ ($n \geq 3$) となる数列 $\{a_n\}$ である。

ある正の整数 m が与えられたとき、 a_m の値を計算するには、 a_1, a_2, a_3, \dots と順に計算していくのが簡単であるが、ここでは敢えて a_m の定義式を展開していく方法で計算する。例えば、 $m = 5$ の場合、 a_5 の定義から $a_5 = a_4 + a_3$ である。次に a_3 を計算することになると、 a_3 の定義から $a_3 = a_2 + a_1$ であり、 $a_1 = a_2 = 1$ であるから、 $a_3 = 2$ である。同様にして $a_4 = 3$ であることが計算できるので、 $a_5 = 3 + 2 = 5$ であることがわかる。

この方式をアルゴリズムの形で書くには、次のことに注意しなければならない。例えば、 a_5 の計算には a_3 と a_4 の両方を計算することが必要であるが、そのためには、 a_3 の計算をしている間は「 a_4 の計算が必要」ということを別の場所に保存しておき、 a_3 の計算が終わったらそれを取り出して a_4 の計算を実行する。また、 a_4 の計算中は、先に計算した a_3 の結果はどこか別の場所に保存する必要がある。

そのような保存場所が何個必要になるかは前もってわからないので、スタックというデータの記憶方式を使用する。

- スタック $S = \{i, s_1, s_2, s_3, \dots\}$ は、無限個の変数 s_1, s_2, s_3, \dots と変数 i が組になったものである。
- i の値は 0 以上の整数である。 $i = 0$ の場合は何もデータが保存されていないことを意味し、 $i > 0$ の場合は、 s_1, \dots, s_i に何らかのデータが保存されている。
- S にデータ x を保存するには、 i の値を 1 増やし、 s_i の値を x にする。
- S からデータを取り出すには、 s_i の値を取り出したデータとし、 i の値を 1 減らす。

スタック $V = \{j, v_1, v_2, v_3, \dots\}$ についても同様である。

S は、これから行うべき計算は何かということを記憶し、そこに保存するデータは、整数の他に '+' という記号も許されるものとする。 S から取り出したデータが整数 n の場合は「 a_n の計算をする」という意味で、計算結果は V に保存する。 S から取り出したデータが記号 '+' の場合は「 V からデータを取り出す操作を 2 回行い、それを足す」という意味で、足した結果は再び V に保存する。

例えば、 a_4 を計算する時は、 $a_3 + a_2$ の計算が必要なので、 S に '+', 3, 2 という値を保存する（表の 1 行目）。次に、 S からデータを 1 個取り出して、それが示す計算を実行する。この場合は 2 が取り出されるので、 a_2 の計算であるが、それは定義から 1 である。この計算結果は V に保存する（表の 2 行

目)。これで a_2 の計算は終わったので、再び S からデータを 1 個取り出す。今度は 3 が取り出されるので、 $a_3 = a_2 + a_1$ の計算が必要になり、 S に '+', 2, 1 という値を保存する（表の 3 行目）。再び S からデータを 1 個取り出すと 1 なので、 V に $a_1 = 1$ を保存する（表の 4 行目）。同様に S から 2 を取り出し、 V に 1 を保存する（表の 5 行目）。次に S から取り出されるのは '+' なので、 V からデータを 2 個取り出し、その足し算をする。結果は、また V に保存する（表の 6 行目）。最後に S から取り出されるのは '+' なので、 V から 2 個取り出し、足し算の結果を V に保存する（表の 7 行目）。

s_1	s_2	s_3	s_4	v_1	v_2	v_3	v_4
'+'	3	2					
'+'	3			1			
'+'	'+'	2	1	1			
'+'	'+'	2		1	1		
'+'	'+'			1	1	1	
'+'				1	2		
				3			

(ア) 次の文章の空欄

(73)	(74)
------	------

 ～

(91)	(92)
------	------

 に当てはまる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。

整数 $m \geq 3$ を入力とし、上の方式で a_m の値を計算して出力するアルゴリズムは次のようになる。

$\{i, s_1, s_2, s_3, \dots\}, \{j, v_1, v_2, v_3, \dots\}$ はスタックである。

s_1 の値を入力された整数 m 、 i の値を 1、 j の値を 0 とする。

$\begin{array}{|c|c|} \hline (73) & (74) \\ \hline \end{array} > 0$ である間、処理 A を繰り返す。

処理 A の始め

もし s_i の値が整数なら処理 B を、記号 '+' なら処理 E を、それぞれ実行する。

処理 B の始め

もし $s_i < 3$ なら処理 C を、そうでなければ処理 D を、それぞれ実行する。

処理 C の始め

$\begin{array}{|c|c|} \hline (75) & (76) \\ \hline \end{array}$ の値を 1 とする。

i の値を $\begin{array}{|c|c|} \hline (77) & (78) \\ \hline \end{array}$ 、 j の値を $\begin{array}{|c|c|} \hline (79) & (80) \\ \hline \end{array}$ とする。

処理 C の終わり

処理 D の始め

s_{i+1} の値を $s_i - 1$ 、 s_{i+2} の値を $\begin{array}{|c|c|} \hline (81) & (82) \\ \hline \end{array}$ とする。

$\begin{array}{|c|c|} \hline (83) & (84) \\ \hline \end{array}$ の値を '+' とする。

i の値を $\begin{array}{|c|c|} \hline (85) & (86) \\ \hline \end{array}$ とする。

処理 D の終わり

処理 B の終わり

処理 E の始め

v_{j-1} の値を $v_{j-1} + \begin{array}{|c|c|} \hline (87) & (88) \\ \hline \end{array}$ とする。

i の値を $\begin{array}{|c|c|} \hline (89) & (90) \\ \hline \end{array}$ 、 j の値を $\begin{array}{|c|c|} \hline (91) & (92) \\ \hline \end{array}$ とする。

処理 E の終わり

処理 A の終わり

v_1 の値を出力する。

【

(73)	(74)
------	------

 ～

(91)	(92)
------	------

 の選択肢】

- (11) $i - 2$ (12) $i - 1$ (13) i (14) $i + 1$ (15) $i + 2$
 (16) $j - 2$ (17) $j - 1$ (18) j (19) $j + 1$ (20) $j + 2$
 (21) $s_i - 2$ (22) $s_i - 1$ (23) s_i (24) $s_i + 1$ (25) $s_i + 2$
 (26) v_{j-2} (27) v_{j-1} (28) v_j (29) v_{j+1} (30) v_{j+2}
 (31) 0 (32) 1 (33) m

(イ) 次の文章の空欄

(93)	(94)
------	------

 ～

(99)	(100)
------	-------

 に入るもっとも適切な数字を解答欄にマークしなさい。

- 処理 A の実行回数は、 $m = 5$ の場合は

(93)	(94)
------	------

 回、 $m = 8$ の場合は

(95)	(96)
------	------

 回である。
- アルゴリズム実行中の i の最大値は、 $m = 5$ の場合は

(97)	(98)
------	------

、 $m = 8$ の場合は

(99)	(100)
------	-------

 である。