〔C1
C2〕−2023−

# デジタル

## 専門(記述式)試験問題

注 意 事 項

**1**. 問題は**6題**あります。このうち**任意の2題**を選んで解答してください。なお、この問題集の**裏表紙**に科目別構成の詳細が記載されていますので、解答開始前によく読んでおいてください。

**2**. 解答時間は**3時間30分**です。

**3**. 答案用紙の記入について
㈠ **答案用紙は表面のみを使用してください**。裏面は採点されません。
㈡ 答案は濃くはっきり書き、書き損じた場合は、解答の内容がはっきり分かるように訂正してください。
㈢ 答案用紙は**4枚つづり2冊**です。問題**1題につき1冊**を使用してください。
㈣ 答案用紙の表紙の各欄にそれぞれ必要事項を記入してください。
問題番号欄には、解答した問題の番号を記入してください。
㈤ 答案用紙各枚の右上にある〔No.　　〕の欄には問題番号を、(　枚目)の欄には解答した問題ごとに何枚目の答案かを記入してください。
㈥ 試験の公正を害するおそれがありますので、答案用紙の氏名欄以外に氏名その他解答と関係のない事項を記載しないでください。

**4**. 下書き用紙はこの問題集の**中央部**にとじ込んであります。**試験官の指示**に従って、**試験開始後に**問題集から下書き用紙だけを慎重に**引きはがして**使用してください。なお、誤って問題集を破損しても、問題集の交換はできませんので注意してください。

**5**. この問題集で単位の明示されていない量については、全て国際単位系(SI)を用いることとします。

**6**. この問題集は、本試験種目終了後に持ち帰りができます。

**7**. 本試験種目の途中で退室する場合は、退室時の問題集及び下書き用紙の持ち帰りはできませんが、希望する方には後ほど渡します。別途試験官の指示に従ってください。なお、試験時間中に、この問題集から**下書き用紙以外**を切り取ったり、問題を転記したりしないでください。

**8**. 下欄に受験番号等を記入してください。

| 第1次試験地 | 試験の区分 | 受験番号 | 氏　　名 |
|---|---|---|---|
| | デジタル | | |

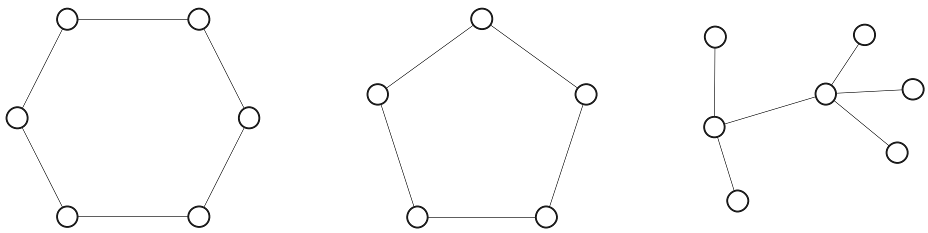
**指示があるまで中を開いてはいけません。**

**途中で退室する場合**………本試験種目終了後の問題集の持ち帰りを　**希望しない**

【No.　1】 以下の設問に答えよ。

$x$ 以上 $y$ 以下の整数の集合を $[x, y]$ で表すとする。また、本問にて扱われる全てのグラフは単純な無向グラフであるとする。$k$ を正整数とする。グラフ $G = (V, E)$ の各頂点に対して $[0, k-1]$ 中の値を割り当てる関数 $f: V \to [0, k-1]$ で、任意の $\{u, v\} \in E$ について $f(u) \neq f(v)$ が成り立つようなものをグラフ $G$ の $k$–彩色と呼ぶ。また、そのような関数 $f$ が存在するとき、$G$ は $k$–彩色可能であると呼ぶ。$G$ が $k$–彩色可能であるような最小の $k$ を $G$ の彩色数と呼び、$\chi(G)$ で表すとする。グラフ $G$ の最大次数を $\Delta(G)$ で表すとする。

(1)　次に示す三つのグラフ $G_1$，$G_2$，$G_3$ の彩色数 $\chi(G_1)$，$\chi(G_2)$，$\chi(G_3)$ を求めよ。



$G_1$　　　$G_2$　　　$G_3$

(2)　任意のグラフ $G = (V, E)$ の彩色数 $\chi(G)$ が $\chi(G) \leqq \Delta(G) + 1$ を満たすことは、次に示すような手続きにより $G$ の $(\Delta(G)+1)$–彩色 $f: V \to [0, \Delta(G)]$ が得られることから分かる。この手続きは $f(v)$ が全ての $v \in V$ に対して未定義である状態より開始し、各頂点 $v$ に対する $f(v)$ の値を順次決定していくことで最終的に $(\Delta(G)+1)$–彩色 $f$ を得る。

1: $v_0, v_1, v_2, \ldots, v_{n-1}$ を $V$ 中の頂点全てを任意の順番で並べた列とする
2: $N_i$ を頂点 $v_i$ の隣接頂点集合とする
3: for $i = 0$ to $n-1$ do
4: 　　$W_i = N_i \cap \{v_j \mid 0 \leqq j < i\}$ とする
5: 　　$Z = [0, \Delta(G)] \setminus \{f(v) \mid v \in W_i\}$ とする（$\setminus$ は差集合演算を表す）
6: 　　任意の値 $x \in Z$ を選び $f(v_i) = x$ と定める
7: end for

上の手続きの 6 行目において、ある $x$ を必ず選ぶことができる、すなわち、$Z$ が空でないことは保証されている。その理由を説明せよ。

(3)　$\chi(G) = \Delta(G) + 1$ が成立するような頂点数が 4 のグラフ $G$ を一つ図示せよ。

(4) $\chi(G)$ と $\Delta(G)+1$ の差が最も大きくなるような頂点数が 6 のグラフ $G$ を一つ図示せよ。

(5) $P=\{p_0, p_1, \ldots, p_{n-1}\}$ を 2 次元平面 $\mathbb{R}^2$ 上の相異なる $n$ 個の点から成る集合とする。点 $q \in \mathbb{R}^2$, $r \in \mathbb{R}^2$ に対して、$q, r$ 間の距離（2 次元ユークリッド距離）を $\|q, r\|$ で表すものとする。$P$ 及び正の実数 $\ell > 0$ に対して、グラフ $G_\ell(P) = (V_\ell(P), E_\ell(P))$ を次のように定義する。

- $V_\ell(P) = P$
- $\{p_i, p_j\} \in E_\ell(P) \Leftrightarrow \|p_i, p_j\| \leqq \ell \ (0 \leqq i < n,\ 0 \leqq j < n,\ i \neq j)$

このとき、以下の問いに答えよ。

(a) $n=5$ として、$\mathbb{R}^2$ 上の相異なる点集合 $P=\{p_0, p_1, p_2, p_3, p_4\}$ を以下のように定める。

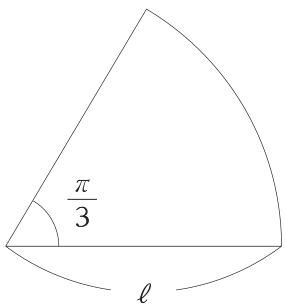
$$p_0=(1,1),\ p_1=(1,3),\ p_2=(2,2),\ p_3=(3,5),\ p_4=(2,4)$$

グラフ $G_2(P)$ を図示せよ。ただし、$P$ 中の各点との対応が分かるように、図示したグラフの頂点には $p_0, p_1, \ldots, p_4$ をラベルとして付記すること。

(b) $\chi(G_1(P))$ が最も大きくなるような $n$ 点の集合 $P$ を $X$ とする。$\chi(G_1(X))$ の値の漸近的評価として最も妥当なものを次の①～⑤から選べ。

① $O(1)$　② $O(\log n)$　③ $O(\sqrt{n})$　④ $O(n)$　⑤ $O(n^2)$

(c) $\ell > 0$ を任意の正の実数とし、$S$ を次に示すような、半径 $\ell$、中心角 $\dfrac{\pi}{3}$〔rad〕の扇形の図形とする。点 $q, r$ を $S$ の内部（境界線上を含む。）の任意の 2 点とするとき、$\|q, r\| \leqq \ell$ が必ず成り立つことを示せ。



$S$

ただし、必要であれば、次に示す加法定理の公式を証明なしに用いてよい（$0 \leqq \theta < 2\pi$, $0 \leqq \varphi < 2\pi$）。

$$\cos(\theta-\varphi)=\cos\theta\cos\varphi+\sin\theta\sin\varphi$$

(d)　三つの頂点から成り、どの異なる 2 頂点間も辺をもつようなグラフを 3–クリークと呼ぶ。$G_\ell(P)$ が $\Delta\bigl(G_\ell(P)\bigr) \geqq 7$ を満たすとき、$G_\ell(P)$ は必ず 3–クリークを部分グラフとして含むことを、(c)で示した事実を用いて示せ。

(e)　$G_\ell(P)$ が 3–クリークを部分グラフとして含まないとき、$\chi\bigl(G_\ell(P)\bigr) \leqq 7$ が成り立つことを示せ。

(6)　$n$ 台の無線基地局が配置された通信システムを考える ($n > 0$)。各基地局は 2 次元平面 $\mathbb{R}^2$ 上の相異なる点とみなし、電波は配置された点を中心とした半径 1 の同心円内(境界を含む。)に到達するものとする。基地局間での電波の干渉を避けるために、電波の到達範囲が重複する二つの基地局には異なる利用周波数帯を割り当てたい。$P = \{p_0, p_1, \ldots, p_{n-1}\}$ を基地局の配置に対応した点集合とするとき、$\chi\bigl(G_2(P)\bigr)$ 個の異なる周波数帯が利用可能であれば、そのような周波数割当てが可能となることを示せ。

【No.　2】 コンピュータにおける乗算に関する以下の設問に答えよ。

Ⅰ．次の符号なし２進数の乗算を２進数で行い、その計算結果を導出過程とともに示せ。

（ア）　$1000 \times 1001$

（イ）　$1101 \times 1111$

（ウ）　$1101 \times 100$

Ⅱ．図Ⅰは、乗算命令を用いずに符号なし２進数の乗算 $M = A \times B$ を行うことを意図した、MIPS アセンブリコードである。このコードについて、以下の問いに答えよ。

なお、$ から始まるオペランドはレジスタを表すとし、$zero は常に値０が格納されている特殊なレジスタである。各命令の意味は右に括弧書きで示している。計算によって発生し得るオーバフローについては、考慮しなくてよいものとする。

| 行番号 | | | | 意味 |
|---|---|---|---|---|
| 1 | | move | $t6, $zero | ($t6 に０を格納) |
| 2 | L0: | andi | $t7, $s2, 1 | ($s2 の値と即値１との論理積を $t7 に格納) |
| 3 | | beq | $t7, $zero, L1 | ($t7 の値が０と等しいならば L1 に分岐) |
| 4 | | add | $t6, $t6, $s1 | ($t6 に $s1 の値を加算) |
| 5 | L1: | srl | $s2, $s2, 1 | ($s2 の値を１ビット右論理シフト) |
| 6 | | sll | $s1, $s1, 1 | ($s1 の値を１ビット左論理シフト) |
| 7 | | bne | $s2, $zero, L0 | ($s2 の値が０と等しくなければ L0 に分岐) |

図Ⅰ　符号なし２進数の乗算を行うコード

(1)　図Ⅰのコードにおいて、被乗数 $A$、乗数 $B$、乗算結果 $M$ を格納する用途に用いられているレジスタはどれか、それぞれ示せ。

(2)　被乗数 $A$ 及び乗数 $B$ が次の(ア)、(イ)、(ウ)に示した組合せである場合に、コード中２行目の andi 命令が実行される回数及び４行目の add 命令が実行される回数を、それぞれ求めよ。

（ア）　$A = 0000\ 0111\ 1110\ 0111,\ B = 0011\ 1111\ 1111\ 1110$

（イ）　$A = 0000\ 0111\ 1110\ 0111,\ B = 0101\ 0101\ 0101\ 0101$

（ウ）　$A = 0100\ 1010\ 0100\ 1001,\ B = 0001\ 1110\ 0111\ 1100$

Ⅲ． 2 進数表記した乗数に、値が 1 であるような桁が多く含まれると、乗算の計算に多くの加算が必要となる。この加算を、値が 1 である桁が多数連続している場合に着目して削減したい。

ここで、例えばⅡ⑵の(ア)の $B = 0011\ 1111\ 1111\ 1110$ に対し、$B$ の中で連続する 1 の最左端の一つ左の桁のみが 1 であるような $L = 0100\ 0000\ 0000\ 0000$ 及び $B$ の中で連続する 1 の最右端と同じ桁のみが 1 であるような $R = 0000\ 0000\ 0000\ 0010$ という二つの値を定義すると、$B = L - R$ が成り立つことから、$A \times B = (A \times L) - (A \times R)$ が成り立つ。これを一般化すると、$n$ 桁の 2 進数 $B = b_n b_{n-1} \cdots b_2 b_1$ $(b_i \in \{0, 1\},\ n \geqq i \geqq 1)$ が、値が 1 である桁の連続を一つだけ含み、その両端が $x$ 桁目及び $y$ 桁目 $(n > x \geqq y \geqq 1)$ であるとき(上記 $B = 0011\ 1111\ 1111\ 1110$ の例では $x = 14,\ y = 2$)、二つの 2 進数 $L = l_n l_{n-1} \cdots l_2 l_1$, $R = r_n r_{n-1} \cdots r_2 r_1$ を、

- 1 の連続部分左端にあたる $b_{x+1} b_x = 01$ に対して $l_{x+1} = 1$
- 1 の連続部分右端にあたる $b_y b_{y-1} = 10$ に対して $r_y = 1$ (ただし、$b_0 = 0$ とみなし、$y = 1$ の場合も $r_y = 1$ とする。)
- 上記以外の桁では $l_i = r_i = 0$

となるよう定めると、$B = L - R$ が成り立つため、$A \times B$ は $L$ と $R$ を用いて $(A \times L) - (A \times R)$ と書ける。

また、$B$ が、値が 1 である桁の連続を $m$ 個含む場合、そのうちの一つの連続部分のみを含むような $m$ 個の 2 進数 $B_1, B_2, \ldots, B_m$ (ここで $B = B_1 + B_2 + \cdots + B_m$) が定義でき、各 $B_j$ $(1 \leqq j \leqq m)$ に対して $L_j, R_j$ を定めることができるため

$$B = L_1 - R_1 + L_2 - R_2 + \cdots + L_m - R_m \quad \cdots\cdots①$$

が成り立つ。

以上を踏まえ以下の問いに答えよ。

⑴ Ⅱ⑵の(イ)、(ウ)で示した $B$ を、式①の形にそれぞれ展開したものを示せ。

⑵ 図Ⅰのコードを改変したものを、図Ⅱに示す。図Ⅱのコードが正しく乗算を行えるように、㋐～㋓に適切な語句を入れよ。

なお、㋐には即値が、㋑、㋒、㋓には分岐先ラベルが入る。

⑶ Ⅱ⑵の(ア)、(イ)、(ウ)で示した $A,\ B$ の組合せに対し、図Ⅱのコードを用いて $A \times B$ を計算したとき、11 行目の add 命令の実行回数と 13 行目の sub 命令の実行回数の合計は、いくつになるかそれぞれ求めよ。

| 行番号 | | | | 意味 |
|---|---|---|---|---|
| 1 | | move | $t6, $zero | |
| 2 | | li | $t1, 1 | ($t1 に即値 1 を格納) |
| 3 | | li | $t2, 2 | ($t2 に即値 2 を格納) |
| 4 | | li | $t3, 3 | ($t3 に即値 3 を格納) |
| 5 | | sll | $s2, $s2, 1 | |
| 6 | L2: | andi | $t7, $s2, ㋐ | |
| 7 | | beq | $t7, $zero, L5 | |
| 8 | | beq | $t7, $t1, ㋑ | |
| 9 | | beq | $t7, $t3, ㋒ | |
| 10 | | beq | $t7, $t2, ㋓ | |
| 11 | L3: | add | $t6, $t6, $s1 | |
| 12 | | j | L5 | (L5 に無条件分岐) |
| 13 | L4: | sub | $t6, $t6, $s1 | ($t6 から $s1 の値を減算) |
| 14 | L5: | srl | $s2, $s2, 1 | |
| 15 | | sll | $s1, $s1, 1 | |
| 16 | | bne | $s2, $zero, L2 | |

図Ⅱ　改変されたコード

【No.　3】 以下の設問に答えよ。

デジタルデータの伝送や保存の際に発生するデータ誤りを検出するために、誤り検出符号が用いられる。現在広く用いられている誤り検出符号の一つにCRC(Cyclic Redundancy Check：巡回冗長検査)がある。

誤り検出符号では、送信する情報ビット列から検査ビット列を計算し、元の情報ビット列に検査ビット列を付加した符号ビット列が送信される。受信時には、受け取った符号ビット列中の情報ビット列を元に検査ビット列を改めて計算し直し、符号ビット列に付加されてきた検査ビット列と比較することによって、受け取ったデータの誤りを検出する。

CRCの場合、送信する情報ビット列の各ビットを係数とする多項式 $F(x)$ を作り、あらかじめ決めてある生成多項式 $G(x)$ で除算し、得られた余りの多項式の係数を検査ビット列として用いる。ただし、多項式の係数の値は0又は1に限られ、全ての計算は2の剰余系(mod 2)で行われる。

例えば、生成多項式 $G(x) = x^4 + x^2 + x + 1$ を用いて、3ビットの情報ビット列100を符号化する場合、まず、情報ビット列を係数とする多項式 $1 \times x^2 + 0 \times x + 0$ を作り、これに $x$ を生成多項式の次数乗した $x^4$ を掛けて、$F(x) = x^6$ を作る。この多項式を以下のように、$G(x)$ で割って余りの多項式を求める。

$$
\begin{array}{r|ccccccc}
 & & & & & x^2 & & +1 \\ \cline{2-8}
x^4 \qquad + x^2 + x + 1 & x^6 & & & & & & \\
 & x^6 & & +x^4 & +x^3 & +x^2 & & \\ \cline{2-6}
 & & & x^4 & +x^3 & +x^2 & & \\
 & & & x^4 & & +x^2 & +x & +1 \\ \cline{4-8}
 & & & & x^3 & & +x & +1
\end{array}
$$

すなわち、$x^6 = (x^4 + x^2 + x + 1)(x^2 + 1) + x^3 + x + 1$ となり、検査ビット列1011が得られ、符号ビット列は1001011となる。ただし、係数の計算は2の剰余系(mod 2)で行われているので注意すること。

上記の割り算は、係数だけを取り出して、より簡略に以下のように記述することもできる。

$$
\begin{array}{r|ccccccc}
 & & & & & 1 & 0 & 1 \\ \cline{2-8}
10111 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 0 & 1 & 1 & 1 & & \\ \cline{2-6}
 & & 0 & 1 & 1 & 1 & 0 & \\
 & & 0 & 0 & 0 & 0 & 0 & \\ \cline{3-7}
 & & & 1 & 1 & 1 & 0 & 0 \\
 & & & 1 & 0 & 1 & 1 & 1 \\ \cline{4-8}
 & & & & 1 & 0 & 1 & 1
\end{array}
$$

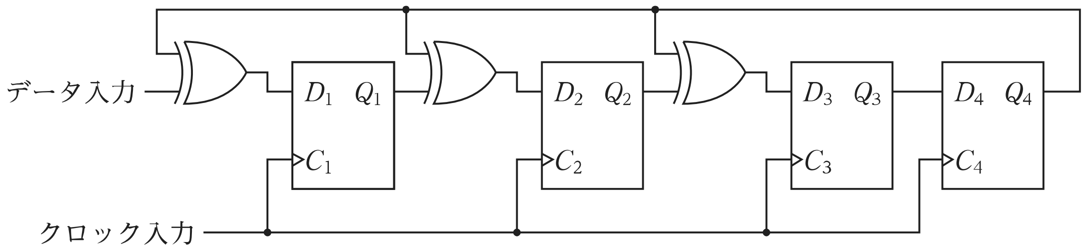
引き算に相当する部分で、2の剰余系(mod 2)で計算をしているため、ビットごとのXOR(排他的論理和)に置き換わっている点に注意すること。

⑴　上の例で得られた符号ビット列 1001011 を係数とする多項式を、生成多項式 $G(x) = x^4 + x^2 + x + 1$ で割ったときの余りを求めよ。

⑵　生成多項式 $G(x) = x^4 + x^2 + x + 1$ を用いて、3 ビットの情報ビット列 111 を符号化せよ。

⑶　同様にして、000 から 111 まで全ての 3 ビット情報ビット列に対して、生成多項式 $G(x) = x^4 + x^2 + x + 1$ を用いた符号ビット列を求め、それぞれの符号ビット列同士のハミング距離(食い違うビットの数)が分かる表を作れ。また、表よりどのようなことが分かるか、説明せよ。

⑷　以下のような論理回路を考える。




この回路にクロックパルスを 7 回入力し、クロックパルスの立ち上がりの度に、データ入力にビット列 1000000 中の各ビットの値を左から順に与えていくものとする。このとき四つの D フリップフロップの出力 $Q_1$～$Q_4$ はどのように変化し、最終的にどのような値となるかを示せ。

ただし、D フリップフロップの初期状態は $Q_1 = Q_2 = Q_3 = Q_4 = 0$ であるものとする。

⑸　⑷に示された回路に、データ入力としてビット列 1110000 中の各ビットの値を左から順に与えていった場合には、四つの D フリップフロップの出力 $Q_1$～$Q_4$ はどのように変化し、最終的にどのような値となるかを示せ。

⑹　⑷に示された回路について考える。以下の問いに答えよ。

(a)　⑷、⑸より、この回路はどのような働きをする回路と考えられるか、説明せよ。

(b)　三つの XOR ゲートと 4 ビットの D フリップフロップによるシフトは、行われる一連の計算の流れの中で、具体的にどのような役割を果たしているのか、説明せよ。

(c)　右端の D フリップフロップの出力 $Q_4$ はどのような意味をもつ信号であり、なぜ三つの XOR ゲートに入力されるのか、説明せよ。

(d)　右から 2 番目の D フリップフロップの出力 $Q_3$ と右端の D フリップフロップの入力 $D_4$ の間に XOR ゲートが必要ないのはなぜか、説明せよ。

【No.　4】 以下の設問に答えよ。

最短経路問題とは、グラフにおいてある頂点から別の頂点へ至る際に、通過する辺の重みの和(経路長)を最小化する問題である。最短経路問題を解くアルゴリズムとして、例えばダイクストラ法、ベルマン・フォード法が知られている。前者では辺の重みは負でない値に限られるが、後者では負の重みの辺を含むことも許される。

本問のＣ言語プログラムにおいて、グラフは隣接行列で表現し、２次元配列で表される。例えば、隣接行列 graph について、頂点 i から j への重みは graph[i][j] であり、接続されていない場合はこの値は０である。なお、本問のプログラムでは重みが０である辺をもつグラフは考えない。プログラムでは <limits.h> が読み込まれており、INT_MAX は int で表現できる最大値である(無限大として使用している。)。また、<stdlib.h> が読み込まれており、malloc 関数は失敗しないものとする。また、真偽値を表す記号定数 TRUE 及び FALSE を次のように定義しておく。

```
#define TRUE !0
#define FALSE 0
```

本問において、計算される経路長は INT_MAX 未満であることが保証されているものとする。

(1)　ダイクストラ法では、辺の重みが全て負でないグラフについて、起点とする頂点を $v_0$ としたとき、その他の頂点への最短経路長を次のように求める。

1. $v_0$ への暫定経路長を０、$v_0$ 以外の頂点への暫定経路長を無限大とする。頂点集合 $U = \{v_0\}$ とする。
2. $U$ 内の頂点に隣接していて、$U$ に含まれない頂点の集合を $W$ とする。$W$ から取り出したそれぞれの頂点 $u$ について、$u$ までの暫定経路長を、「$U$ 内の頂点 $v$ までの暫定経路長と $v$ から $u$ に(接続されている場合)至る辺の重みとの和」の中で最小のものにする。
3. $W$ の中で最小の暫定経路長をもつ頂点 $u$ を選び出し、$u$ を $U$ に追加する。
4. $U$ がグラフの全ての頂点を含むようになるまで、２. に戻って繰り返す。繰返しが終了したとき、各頂点への暫定経路長が最短経路長になっている。

ダイクストラ法で最短経路問題を解くプログラムを、次のようなＣ言語の関数で記述した。ここで、グラフの頂点数は nodes であり、隣接行列は graph である。この関数は頂点番号 start からの最短経路長を配列 dist に計算する。

```
void dijkstra(int *dist, int **graph, int nodes, int start) {
  int *visited = (int *)malloc(sizeof(int) * nodes);
```

```
    int i, j, m;

    for (i = 0; i < nodes; i++) {
      dist[i] = INT_MAX; visited[i] = FALSE;
    }
    dist[start] = 0;
    for (i = 0; i < nodes; i++) {
      [ ㋐ ] ;
      for (j = 0; j < nodes; j++) {
        if ( [ ㋑ ] && [ ㋒ ] < dist[j]) dist[j] = [ ㋒ ] ;
      }
      m = INT_MAX;
      for (j = 0; j < nodes; j++) {
        if (visited[j] == FALSE && dist[j] < m) {
          m = dist[j]; [ ㋓ ] ;
        }
      }
    }
    free(visited);
  }
```

(a) 図 I のグラフについて、起点を頂点 A として、ダイクストラ法を用いて各頂点への最短経路長を求めたい。表 I の続きを作成して各頂点への暫定経路長を更新し、最短経路長を求めよ。

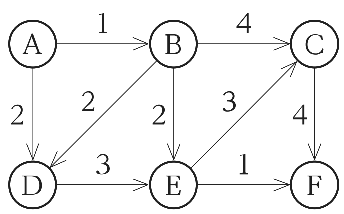



図 I

表 I

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 0 | 1 | ∞ | ∞ | ∞ | ∞ |
| 0 | 1 | ∞ | 2 | ∞ | ∞ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

(b)　プログラム中の㋐〜㋓に適切な式を入れよ。

ただし、同じ記号には同じ式が入る。

(c)　この関数で最短経路長を求めるとき、その最悪計算量のオーダとして最も妥当なのを次の①〜⑩から選べ。

ただし、頂点数を $n$ とし、当てはまるものが複数あるときは、最も小さいオーダを示すこと。

① $O(1)$　② $O(\log n)$　③ $O(n)$　④ $O(n \log n)$　⑤ $O(n\sqrt{n})$

⑥ $O(n^2)$　⑦ $O(n^2 \log n)$　⑧ $O((n \log n)^2)$　⑨ $O(n^2\sqrt{n})$　⑩ $O(n^3)$

(d)　本問の dijkstra 関数は、負の重みをもつ辺を含むグラフに適用させることができない。この関数に負の重みをもつ辺を含むグラフを与えたとき、どのようなことが起こり得るか、例と共に５行以内で説明せよ。

ただし、例を挙げるに当たって図を用いてもよい(図は解答の行数に含めない。)。

(2)　グラフにおいて、起点と終点を定めて、その最短経路を求めるアルゴリズムに A* 法が知られている。A* 法は、経路探索中に通る頂点に対して、そこから終点までの見込みの重みを用い、その頂点までの暫定経路長と終点までの見込みの重みの和が小さいものから優先的に探索していく。これにより、見込みの重みが妥当に計算されれば、終点に向かって効率的に探索を進められる。その代わり、A* 法は定めた終点までの最短経路を求めるものとなり、ダイクストラ法のように起点から全ての頂点までの最短経路を求めるものではなくなる。A* 法は、終点までの見込みの重みが求めやすい場合有効に機能する。例えば、道路や鉄道の経路探索など、見込みの重みとして地理的な直線距離が考えられるような場合が挙げられる。また、A* 法において、見込みの重みを用いずに探索する方法がダイクストラ法であるとも言える。

(1)で作成した、ダイクストラ法で最短経路を求める dijkstra 関数に基づいて、A* 法で頂点 start から goal までの最短経路長を求めて返す関数 astar を次のように作成した。ここで、頂点 a から b への見込みの重みは int adist(int a, int b) で求められ、この値は A* 法が正しく機能する妥当な見込みの重みであるものとする。また、㋐、㋑には(1)(b)で答えた式が補充されているものとする。

```
int astar(int **graph, int nodes, int start, int goal) {
  int *visited = (int *)malloc(sizeof(int) * nodes);
  int *dist = (int *)malloc(sizeof(int) * nodes);
  int i, j, m, next;

  for (i = 0; i < nodes; i++) {
    dist[i] = INT_MAX; visited[i] = FALSE;
  }
```

（下書き用紙）

（下書き用紙）

（下書き用紙）

```
  dist[start] = 0;
  while (start != goal) {
    ㋐ ;
    for (j = 0; j < nodes; j++) {
      if ( ㋑ && ㋔ < dist[j]) dist[j] = ㋔ ;
    }
    m = INT_MAX;
    for (j = 0; j < nodes; j++) {
      if (visited[j] == FALSE && dist[j] < m) {
        m = dist[j]; ㋕ ;
      }
    }
    dist[next] = ㋖ ; start = next;
  }
  m = dist[goal];
  free(dist); free(visited);
  return m;
}
```

(a) プログラム中の㋔、㋕、㋖に適切な式を入れよ。

ただし、同じ記号には同じ式が入る。

(b) 本問の astar 関数は、特定の条件のとき最短経路を求められず、処理が停止しなくなる。これが発生する条件を3行以内で説明せよ。

(3) ベルマン・フォード法では、有向グラフ中の起点とする頂点を $v_0$ としたとき、その他の頂点への最短経路長を次のように求める。

1. $v_0$ への暫定経路長を0、$v_0$ 以外の頂点への暫定経路長を無限大とする。$k=1$ とする。頂点集合 $V_0=\{v_0\}$ とする。
2. $V_k=\{\}$ とする。$V_{k-1}$ に含まれる頂点全てについて次の手順を実行する。$V_{k-1}$ から取り出した頂点 $v_\mathrm{f}$ に隣接するそれぞれの頂点 $v_\mathrm{t}$ について、$v_\mathrm{f}$ までの暫定経路長に $v_\mathrm{f}$ から $v_\mathrm{t}$ に至る辺の重みを加えた値 $d$ が $v_\mathrm{t}$ への暫定経路長を下回っているときには、$v_\mathrm{t}$ への暫定経路長を $d$ とし、さらに $v_\mathrm{t}$ を $V_k$ に(まだ追加されていなければ)追加する。
3. $V_k$ が空であれば終了。それぞれの頂点への暫定経路長が最短経路長になっている。そうでなければ $k$ に1を加え、2. に戻る。

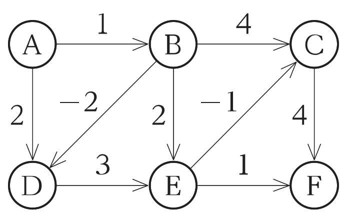
ベルマン・フォード法で最短経路問題を解くプログラムを、次のようなC言語の関数で記述した。引数については(1)のダイクストラ法のプログラムと同様である。ただし、graphの要素の値には負の数が含まれ得る。

この関数は頂点番号startからの最短経路長を配列distに計算する。関数内で計算される経路長は、この関数で最短経路長が正しく計算できる条件下においてはintで表現できるものとする。

```
void bellman_ford(int *dist, int **graph, int nodes, int start) {
  int i, j, endflag;

  for (i = 0; i < nodes; i++) dist[i] = INT_MAX;
  dist[start] = 0;
  do {
    endflag = TRUE;
    for (i = 0; i < nodes; i++) {
      for (j = 0; j < nodes; j++) {
        if ( [ ク ] && dist[i] != INT_MAX && [ ケ ] < dist[j]) {
          dist[j] = [ ケ ] ;
          endflag = FALSE;
        }
      }
    }
  } while (!endflag) ;
}
```

(a) 図Ⅱのグラフについて、起点を頂点Aとして、ベルマン・フォード法を用いて各頂点への最短経路長を求めたい。表Ⅱの続きを作成して各頂点への暫定経路長を更新し、最短経路長を求めよ。




図Ⅱ

表Ⅱ

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 0 | 1 | ∞ | 2 | ∞ | ∞ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

(b)　プログラム中の㋗、㋘に適切な式を入れよ。

ただし、同じ記号には同じ式が入る。

(c)　この関数で最短経路長が正しく計算できるとき、その最悪計算量のオーダとして最も妥当なのを次の①～⑩から選べ。

ただし、頂点数を $n$ とし、当てはまるものが複数あるときは、最も小さいオーダを示すこと。

① $O(1)$　② $O(\log n)$　③ $O(n)$　④ $O(n \log n)$　⑤ $O(n\sqrt{n})$
⑥ $O(n^2)$　⑦ $O(n^2 \log n)$　⑧ $O((n \log n)^2)$　⑨ $O(n^2 \sqrt{n})$　⑩ $O(n^3)$

(d)　本問の bellman_ford 関数は、適切でないグラフを graph に与えられると繰返しが停止しなくなる。繰返しが停止しなくなるのは graph がどのような性質をもつときか、５行以内で説明せよ。

(e)　ベルマン・フォード法は、各頂点に着目すると、頂点 $v$ までの暫定経路長が更新されるとき、「『それまでの頂点 $v$ に隣接する頂点までの暫定経路長に、その頂点から $v$ に至る重みを加えたもの』の中で最小のもの」が求められていることになる。この性質を用いて、ベルマン・フォード法は、ルータのルーティングテーブルを作成する RIP(Routing Information Protocol)に使用されている。いま、ルータ X のルーティングテーブルが表Ⅲのようであったとする。次の(i), (ii), (iii)それぞれの更新情報をルータ X が受け取ったとき、表Ⅲはどのように更新されるか、それぞれ２行以内で説明せよ。

表Ⅲ

| 宛先 | 送出先 | ホップ数 |
|---|---|---|
| 100.?.?.? | ルータ A | 5 |
| 101.?.?.? | ルータ B | 3 |
| 102.?.?.? | ルータ C | 4 |

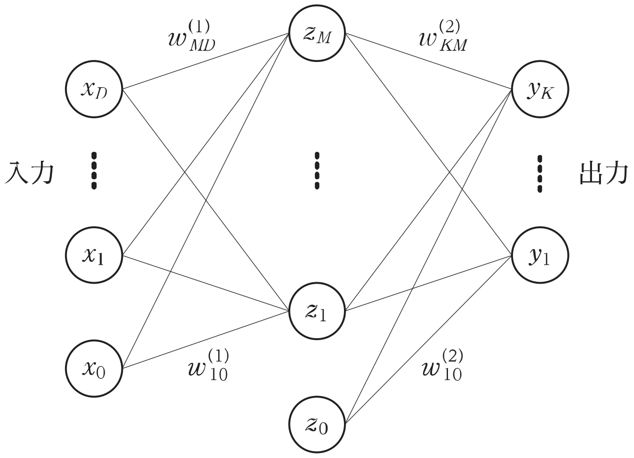
(i)　ルータ B から、ルータ X は 102.?.?.? へホップ数５でたどり着けるという情報を受け取った。

(ii)　ルータ A から、ルータ X は 103.?.?.? へホップ数５でたどり着けるという情報を受け取った。

(iii)　ルータ C から、ルータ X は 100.?.?.? へホップ数４でたどり着けるという情報を受け取った。

【No.　5】 以下の設問に答えよ。

図Ⅰのようなニューラルネットワーク(以下、ネットワークという。)を考える。入力層に入力された $D$ 個の入力 $x_1, x_2, \ldots, x_D$ と定数 $x_0 = 1$ は、バイアスを含めた重み係数 $w_{ji}^{(1)}$ $(0 \leqq i \leqq D,\ 1 \leqq j \leqq M)$ と活性化関数 $h$ により、$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i$，$z_j = h(a_j)$ $(1 \leqq j \leqq M)$ のように中間層の $M$ 個の値 $z_1, z_2, \ldots, z_M$ に変換される。定数 $z_0 = 1$ を含めた $(M+1)$ 個の中間層の値は、さらにバイアスを含めた重み係数 $w_{kj}^{(2)}$ $(0 \leqq j \leqq M,\ 1 \leqq k \leqq K)$ と活性化関数 $\sigma$ により、$b_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j$，$y_k = \sigma(b_k)$ $(1 \leqq k \leqq K)$ のように $K$ 個の出力 $y_1, y_2, \ldots, y_K$ に変換される。




図Ⅰ

(1)　このネットワークのバイアスを含めた全ての重み係数の個数を示せ。

(2)　ネットワークの重み係数を訓練データに適応するように調整することを学習と呼ぶ。ネットワークの教師付き学習は一般に次のような手順で行われる。

ステップ０：$\boxed{\text{㋐}}$ を初期化する。

ステップ１：訓練データの説明変数をネットワークに入力して出力を得る。ネットワークの出力がどの程度目的変数(目標とするネットワークの出力)に近づいているかを $\boxed{\text{㋑}}$ で評価する。

ステップ２：$\boxed{\text{㋒}}$ に関する $\boxed{\text{㋓}}$ の $\boxed{\text{㋔}}$ を求める。

ステップ３：$\boxed{\text{㋕}}$ に基づいて $\boxed{\text{㋖}}$ を更新する。

ステップ４：ステップ１、ステップ２、ステップ３を順に所定の回数繰り返す。

㋐～㋖に当てはまる適切なものを、それぞれ次の語句の中から選び出して入れよ。
ただし、一度使用した語句を再度使用してもよい。
［ 語句：　重み係数、勾配、損失関数 ］

(3) $D=1,\ M=3,\ K=1,\ h(a)=\tanh a,\ \sigma(b)=b$ のとき、以下の問いに答えよ。

(a) $a_j\ (1 \leqq j \leqq M)$ を $x_1$ を用いてそれぞれ表せ。

ただし、総和記号を用いずに展開すること。

(b) $y_1$ を $z_j\ (1 \leqq j \leqq M)$ を用いて表せ。

ただし、総和記号を用いずに展開すること。また、$z_j$ はそれ以上展開しなくてよい。

(c) 入力 $x_1$ をネットワークに入力したときの出力を $y_1$、目的変数を $t$ とする。損失関数を $E_n=\dfrac{1}{2}(y_1-t)^2$ とするとき、$\dfrac{\partial E_n}{\partial w_{ji}^{(1)}}\ (0 \leqq i \leqq 1,\ 1 \leqq j \leqq M)$ と $\dfrac{\partial E_n}{\partial w_{1j}^{(2)}}\ (0 \leqq j \leqq M)$ を求めよ。

ただし、$z_j, y_1$ はそれ以上展開せず使用してよい。また、$h(a)=\tanh a$ の導関数を $h'(a)$ とするとき、$h'(a)=1-h(a)^2$ が成り立つことを用いてよい。

(4) 図Ⅰのネットワークについて、$h(a)=\tanh a,\ \sigma(b)=b$ とする。重み係数を他の重み係数と交換したり重み係数の符号を反転させたりしても、入力に対する出力の関係が変わらない場合がある。重み係数の中に同じ値があるような特殊な場合を除くとき、そのような重み係数の組合せの数を求めよ。

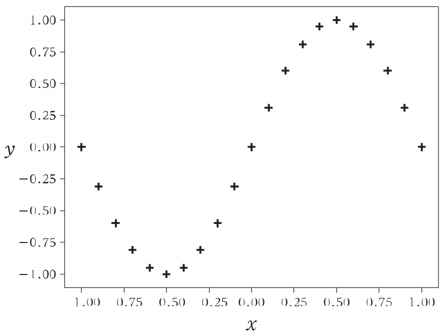
なお、$\tanh(-a)=-\tanh a$ の関係が成り立つ。

(5) ネットワークのバイアスを含めた全ての重み係数をベクトルとして表したものを $\boldsymbol{w}$ とし、損失関数を $E(\boldsymbol{w})$ とする。適当な $\boldsymbol{w}$ の初期値として $\boldsymbol{w}^{(0)}$ から式 $\boldsymbol{w}^{(\tau+1)}=\boldsymbol{w}^{(\tau)}-\eta\nabla E\left(\boldsymbol{w}^{(\tau)}\right)$ により $\boldsymbol{w}$ の更新を繰り返すとき($\tau=0,1,\ldots$)、損失関数 $E(\boldsymbol{w})$ の値がどのように変化するか、更新が停止するのはどのような場合か、また $\eta$ の値によって挙動がどのように変化するかについて合わせて5行程度で説明せよ。

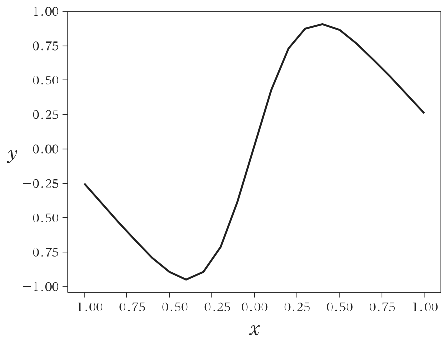
ただし、$\nabla E(\boldsymbol{w})$ は損失関数の勾配、$\eta$ は正の定数とする。

(6) 図Ⅱに表すような、訓練データとして説明変数 $\{X_n\}\ (1 \leqq n \leqq 21)$ ($x$ 軸)と説明変数に対応する目的変数 $\{Y_n\}\ (1 \leqq n \leqq 21)$ ($y$ 軸)を与えて、(5)の損失関数の値を最小化するように $D=1,\ M=3,\ K=1$ のネットワークを学習した。所定の回数、(2)の手続きを繰り返した結果、入力 $x_1$ ($x$ 軸)に対して図Ⅲに示すような値を $y_1$ ($y$ 軸)として出力するようなネットワークが得られた。このネットワークにおける $a_j\ (1 \leqq j \leqq M)$、$z_j\ (1 \leqq j \leqq M)$、$w_{1j}^{(2)}z_j\ (1 \leqq j \leqq M)$ の値($y$ 軸)が入力 $x_1$ ($x$ 軸)に対してどのように変化するかを可視化したものが図Ⅳ、図Ⅴ、図Ⅵのいずれかである。(i) $a_j\ (1 \leqq j \leqq M)$ (ii) $z_j\ (1 \leqq j \leqq M)$ (iii) $w_{1j}^{(2)}z_j\ (1 \leqq j \leqq M)$ と図Ⅳ、図Ⅴ、図Ⅵの対応関係として妥当なのを次の①～⑥から選べ。また、その理由を3行以内で説明せよ。

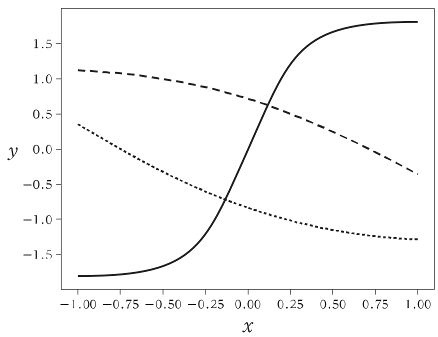
| | (i) | (ii) | (iii) |
|---|---|---|---|
| ① | 図Ⅳ | 図Ⅴ | 図Ⅵ |
| ② | 図Ⅳ | 図Ⅵ | 図Ⅴ |
| ③ | 図Ⅴ | 図Ⅳ | 図Ⅵ |
| ④ | 図Ⅴ | 図Ⅵ | 図Ⅳ |
| ⑤ | 図Ⅵ | 図Ⅳ | 図Ⅴ |
| ⑥ | 図Ⅵ | 図Ⅴ | 図Ⅳ |



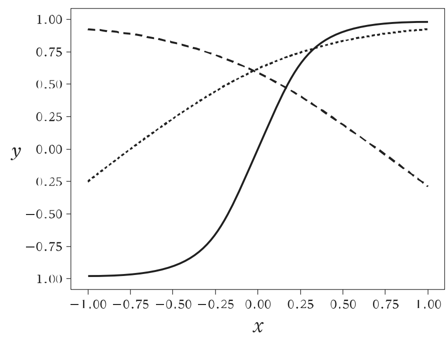

図Ⅱ



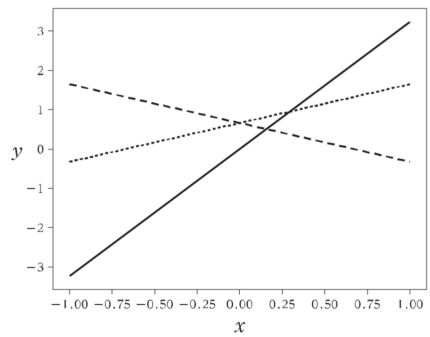

図Ⅲ




図Ⅳ



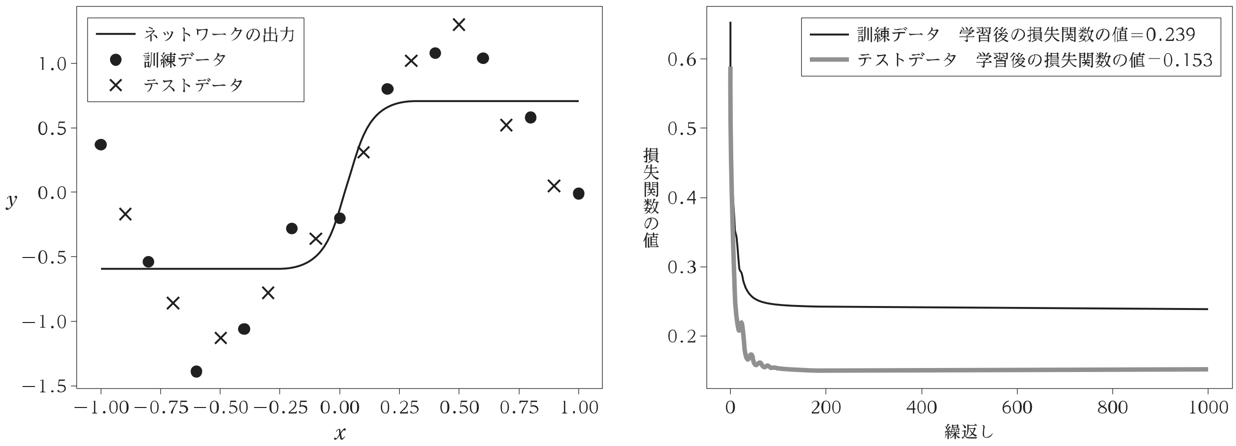

図Ⅴ




図Ⅵ

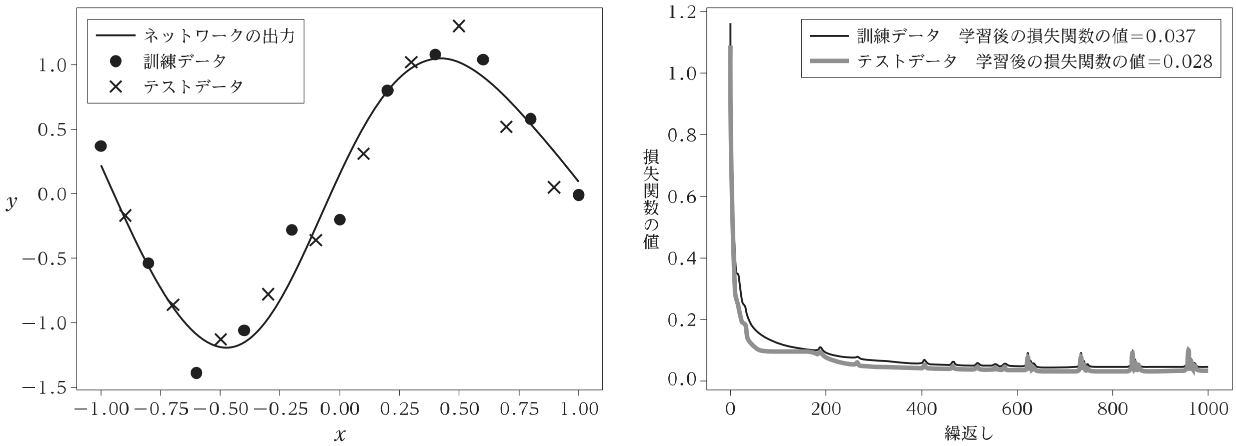
(7) $D=1$, $K=1$ とする。説明変数と目的変数から成る 21 個のデータを 11 個の訓練データと 10 個のテストデータに分割する。$M=1, 2, 10$ の 3 種類のネットワークを訓練データを用いて学習させた結果を図Ⅶ、図Ⅷ、図Ⅸにそれぞれ表す。各図の左図では、訓練データ(●)、テストデータ(×)と共に、(2)の手続きを 1000 回繰り返して得られたネットワークが、入力 $x_1$ ($x$ 軸)に対してどのような値 $y_1$ ($y$ 軸)を出力するかを実線で表している。各図の右図では、繰返しに伴う訓練データとテストデータそれぞれに対する損失関数の値の推移を表している。さらに、右上の値は、学習後の訓練データとテストデータそれぞれに対する損失関数の値を示す。訓練データに含まれないデータに対する推定能力を汎化能力とするとき、この 3 種類のネットワークの中で汎化能力が最も高いものを選び、その理由を 10 行以内で説明せよ。

$M = 1$



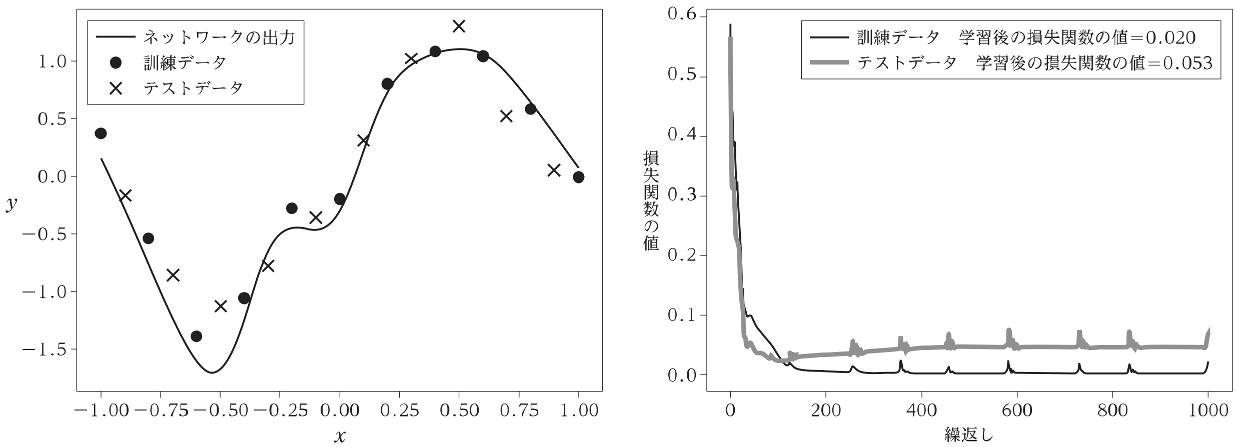

図Ⅶ

$M = 2$




図Ⅷ

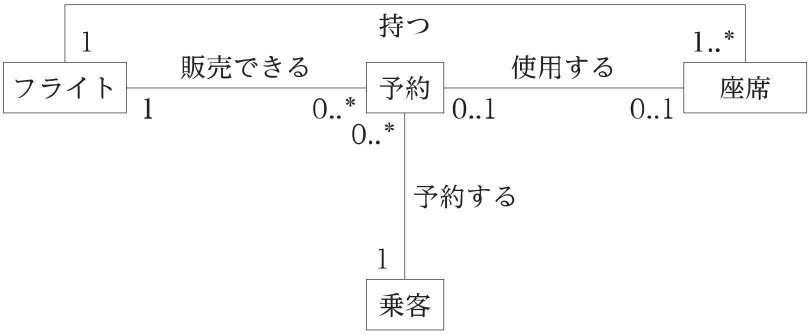
$M = 10$




図Ⅸ

【No.　6】 以下の設問に答えよ。

(1) 図Ⅰは、あるシステムにおけるフライトの座席予約に関するデータをモデリング言語 UML (Unified Modeling Language)を用いて表したものである。




図Ⅰ

(a) 図Ⅰに関する次の記述の㋐～㋓に当てはまる適切なものを、それぞれ次の語句の中から選び出して入れよ。

「図Ⅰは、UML で用いられる図のうち [ ㋐ ] と呼ばれるものである。図Ⅰ中の四角形(「フライト」、「予約」など)は [ ㋑ ] を表し、[ ㋑ ] 間に引かれた線は [ ㋒ ] を表す。また、この線の両端に示される数値に関する表現(「1」や「0..*」など)は [ ㋓ ] を表す。なお、[ ㋓ ] は「下限値..上限値」の形式で表現し、下限値と上限値が同じ場合は、「3」などのようにその値のみを示す。任意数であることは「*」で示す。」

［語句： 実体関連図、コンポーネント図、クラス図、クラス、オブジェクト、実体集合、関連、関係、多重度、集約、利用、属性、集約度、利用数］

(b) 次の記述㋐～㋔のうち、図Ⅰから読み取れることとして妥当なもののみを全て挙げよ。

㋐ 一つのフライトに対して販売できる予約の数は、フライトが持つ座席の数を超えることはない。

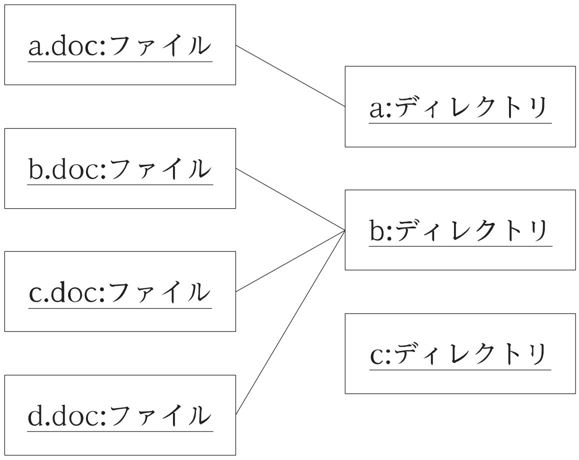
㋑ 予約には、必ず対応するフライトが存在する。

㋒ 予約には、それを予約した乗客が必ず 1 人存在する。

㋓ 乗客は、一つのフライトに対して複数の予約をすることはできない。

㋔ 乗客が予約をした段階で、使用する座席が決まっていなくてもよい。

(2) 図Ⅱは、あるファイルシステム中のファイル群とディレクトリ群について UML のオブジェクト図を用いて表したものである。図Ⅲは、このファイルシステムにおけるファイルとディレクトリの関係を(1)の図Ⅰと同じ図法を用いて表したものである。




図Ⅱ


図Ⅲ

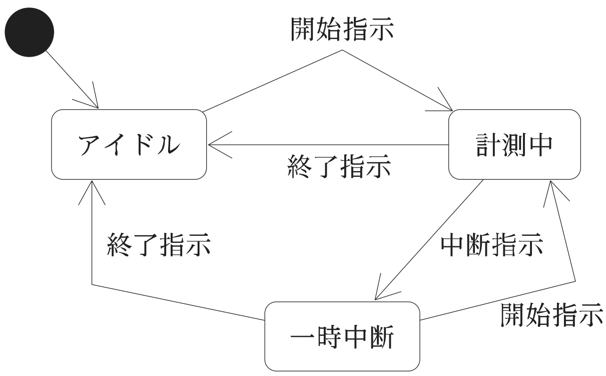
(a)　図Ⅲの㋐、㋑には、(1)(a)の㋓に該当するものが入る。㋐、㋑に当てはまる適切なものを示せ。

(b)　図Ⅱ、図Ⅲでは、ディレクトリ中には複数のファイルが存在し得ることが表されている。実際には、ディレクトリ中にはファイルだけでなくディレクトリも存在し得る。また、ファイル、ディレクトリは、両者に共通の属性として、名前、サイズ、アクセス権をもつ。これらのことも表現できるように図Ⅲを修正したものを描け。

　なお、四角形で示されるもの((1)(a)の㋑)は、図Ⅲに登場するもの以外を追加して用いてもよい。

(c)　図Ⅰ、図Ⅱ、図Ⅲが表す内容に着目して、図Ⅱのようなオブジェクト図と、図Ⅰや図Ⅲのような図((1)(a)の㋐)の類似点及び違いを合わせて5行程度で説明せよ。

(3)　図Ⅳは、あるタイマーの動作をUMLの図のうちの一つを用いて表したものである。




図Ⅳ

(a)　この図の名称を示せ。

(b)　図Ⅳで表されるタイマーの現在の状態が「アイドル」であるとする。この後、「中断指示」、「開始指示」、「中断指示」、「終了指示」の順番でイベントが発生したとき、タイマーの状態がどのように変わるか、タイマーの状態を現在の状態「アイドル」から始めて順番に全て示せ。

(c)　次はある電子レンジの仕様である。この電子レンジの振る舞いを図Ⅳで用いたUMLの図を用いて描け。

なお、状態が並行に変化していることをより直接的に表現すること。

・　加熱する機能をもち、加熱に関しては、加熱をしていない状態「アイドル」と加熱をしている状態「加熱中」をとる。初期状態は「アイドル」である。
・　「アイドル」のときに「スタート」イベントが発生すると「加熱中」になる。
・　「加熱中」のときに「タイムアップ」イベントが発生すると「アイドル」になる。
・　「加熱中」のときに「スタート」イベントが発生すると「ビープ音を鳴らす」ということを実行する。状態は変化しない。
・　「加熱中」のときに「ストップ」イベントが発生すると「アイドル」になる。
・　出力を切り替えることができ、「出力小」、「出力大」の状態をとる。出力に関する状態は、加熱に関する状態と並行に変化する（例えば、加熱に関しては「加熱中」で出力に関しては「出力小」の状態や、加熱に関しては「加熱中」で出力に関しては「出力大」の状態などがあり得る。）。出力に関する初期状態は「出力小」である。
・　出力に関してどの状態にあっても、「切替」イベントでもう一方の状態になる。

(4)　UMLの図のうち、図Ⅰのような図と図Ⅳのような図が表す内容に着目して、これらの図の違いを４行程度で説明せよ。

(5)　ソフトウェア開発の過程において、UMLの図等の形式性をもつ図がよく用いられる。図Ⅰ～図Ⅳはいずれもソフトウェア開発で用いられる図の一例である。このような形式性をもつ図を用いることがソフトウェア開発において有用である理由を６行程度で説明せよ。また、これらの図には様々な種類のものがあるが、複数の図が必要である理由を６行程度で説明せよ。

**科目別構成の詳細**

| 科　　　　目 | 出 題 数 | 問 題 番 号 | ペ ー ジ |
|---|---|---|---|
| 計算機科学 | 1 題 | No. **1** | 1～3 |
| 情報工学（ハードウェア） | 2 題 | No. **2**, **3** | 4～8 |
| 情報工学（ソフトウェア） | 2 題 | No. **4**, **5** | 9～18 |
| 情報技術 | 1 題 | No. **6** | 19～21 |

○　**6 題**のうちから**任意の 2 題**を選んで解答してください。