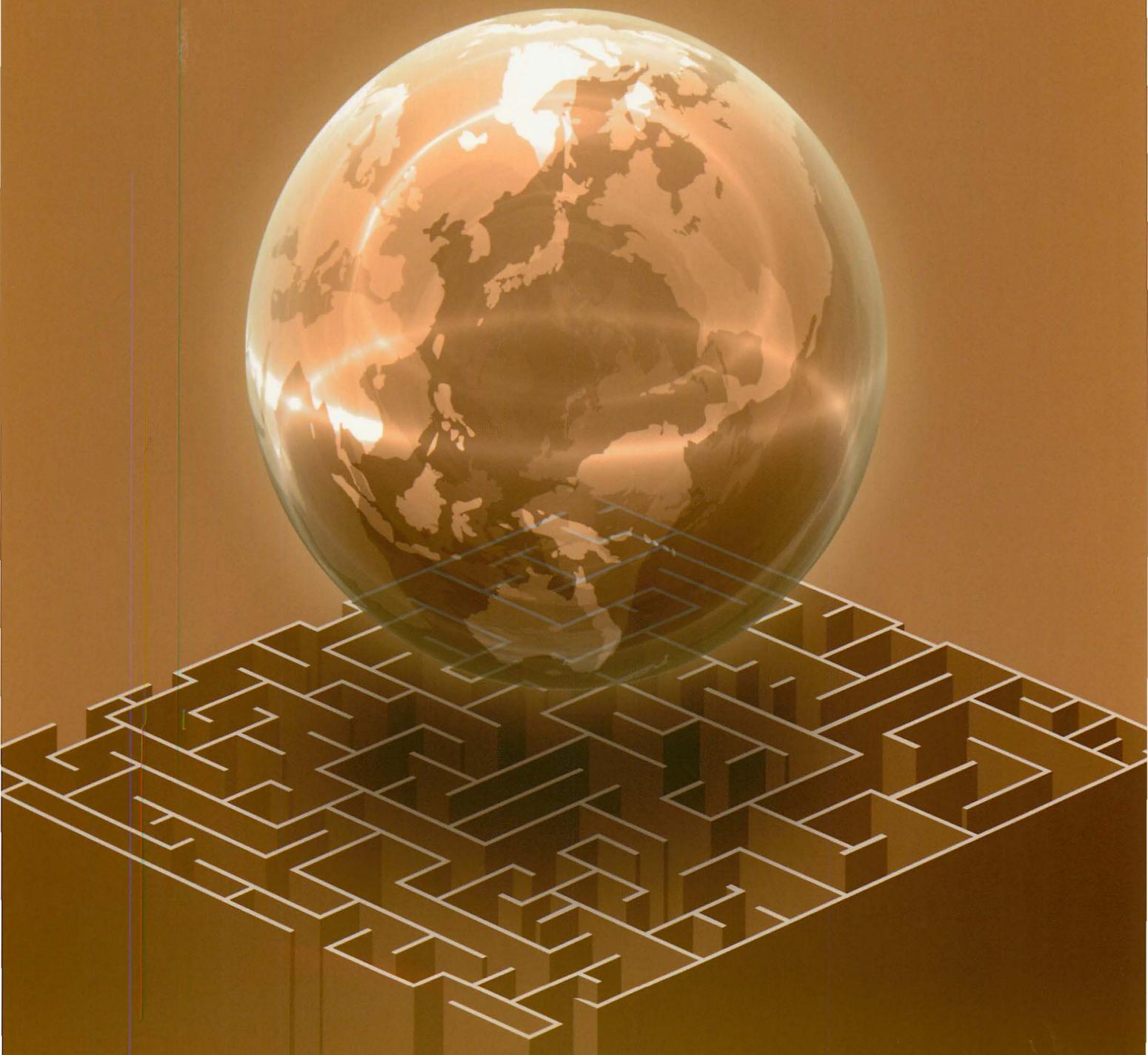


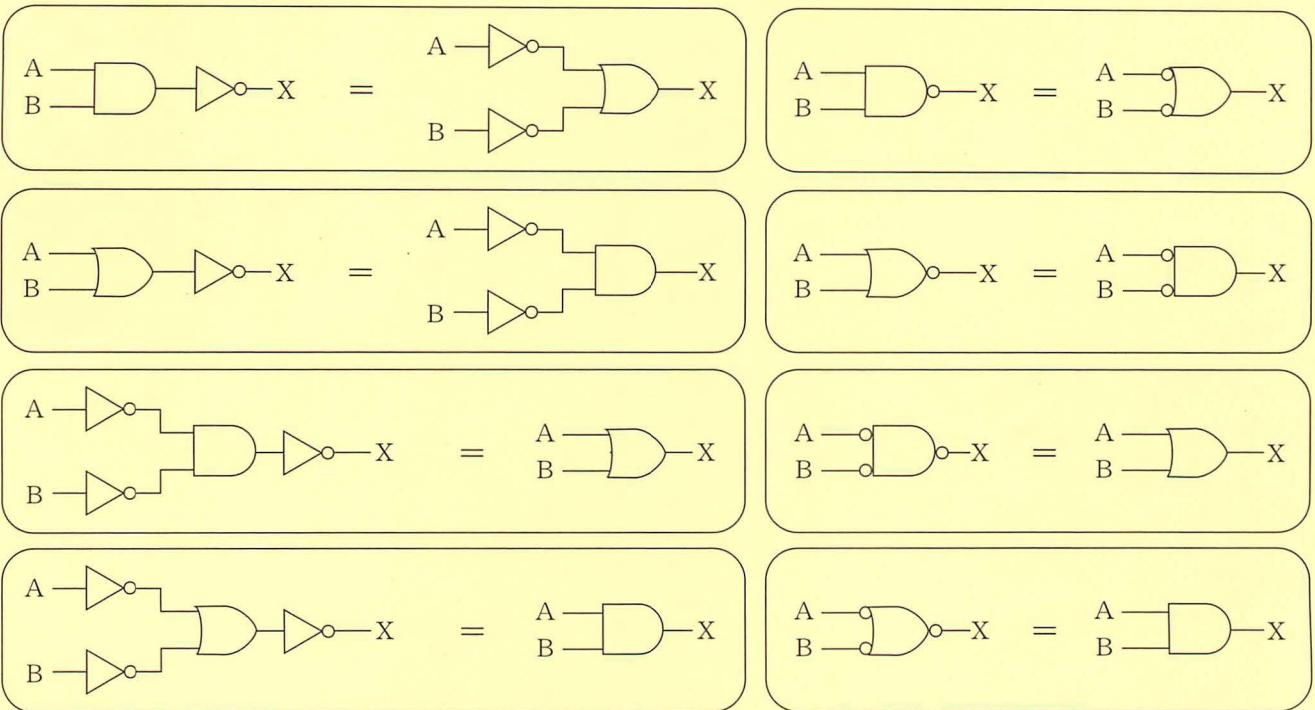
平成23年度版

1級 情報技術検定試験 標準問題集

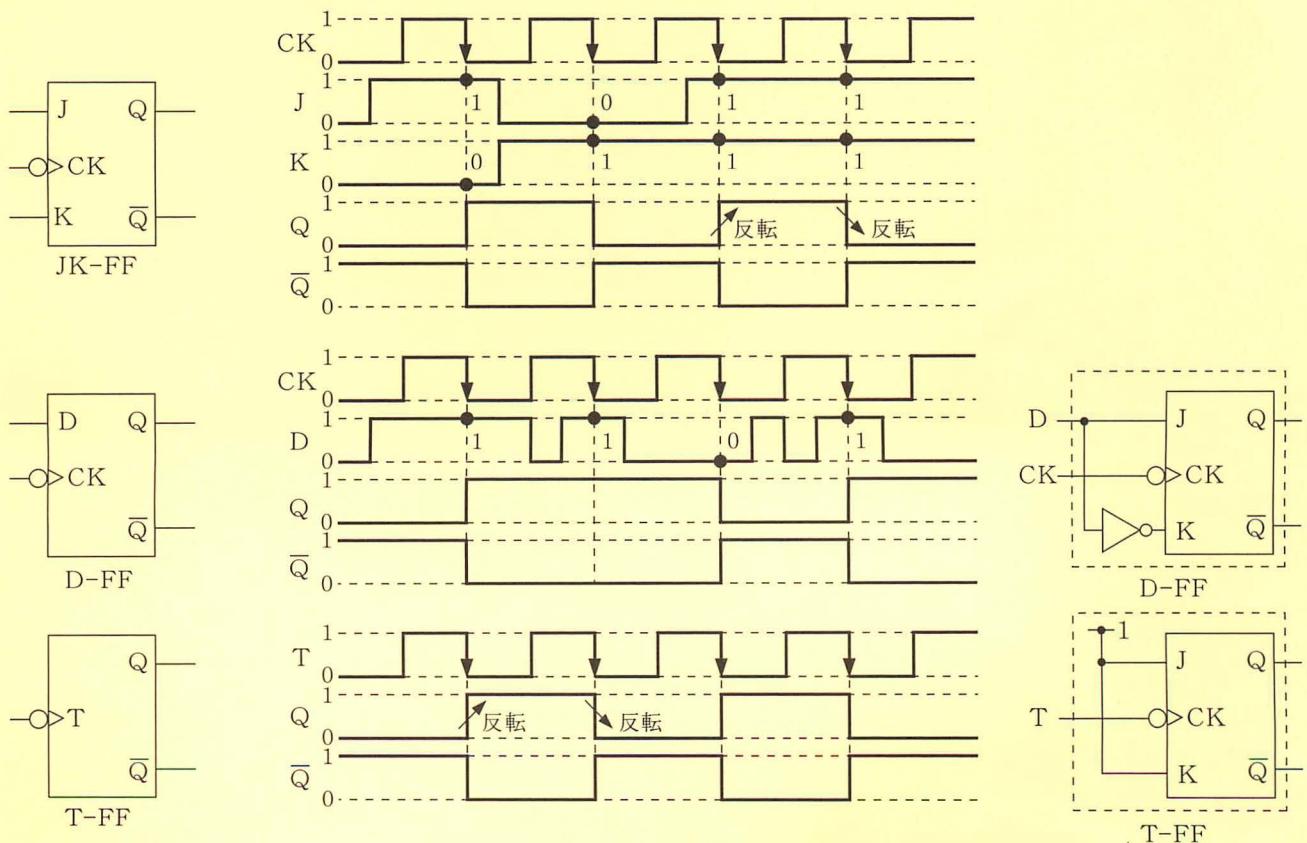


社団法人 全国工業高等学校長協会

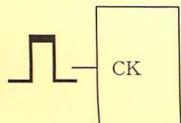
ド・モルガンの定理を利用した論理回路例



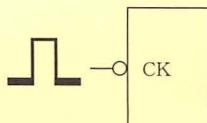
代表的なフリップフロップのタイムチャート



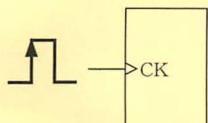
ハイアクティブ



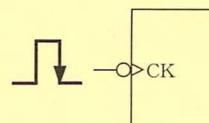
ローアクティブ



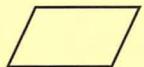
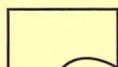
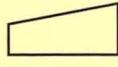
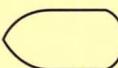
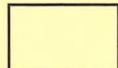
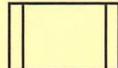
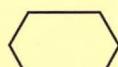
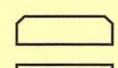
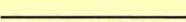
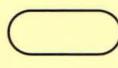
ポジティブエッジ

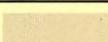
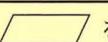


ネガティブエッジ



流れ図の図記号は、J I Sで決められており、おもな図記号を表に示す。

データ		媒体を指定しないデータを示す。 データの入力および出力の時に使用する。
書類		プリンタ出力など人が読める媒体上のデータを示す。
手操作入力		キーボードなど手で操作して入力するデータを示す。
表示		ディスプレイなどに表示するデータを示す。
処理		任意の種類の処理機能を示し、計算や代入などに使用する。
定義済処理		別の場所ですでに定義された一つ以上の処理を示す。
準備		そのあとの動作に影響を与えるための準備を示す。
判断		一つの入り口といくつかの択一的な出口をもち、図記号の中の条件に従って、一つの出口を選ぶ機能を示す。 条件分岐処理・条件による繰り返し処理で使用する。
ループ端		二つの部分からなり、ループのはじまりとおわりを示す。 繰り返し処理で使用する。
結合子		流れ図のほかの場所への出口、またはほかの部分からの入口を示す。
線		図記号をつなぎ、流れを示す。流れの向きを明示するとき、矢印をつける。
端子		プログラムの流れの開始と終了を示す。

※本情報技術検定試験では、の図記号を使い、入出力は、データを使用する。

1級情報技術検定試験

標準問題集

——[I]・[II] C言語 / CASL II ——

社団法人 全国工業高等学校長協会 編

まえがき

本標準問題集は「情報技術検定試験」受検者が、情報技術の知識・技能の習得のための参考になればと考えて編集された参考書である。情報化の進展が著しい今日、コンピュータや情報通信ネットワークなどの情報手段を適切に活用できるような「情報技術に関する資格試験」は多数あり、内容も簡単なものから高度なものまである。それぞれの分野で学習した内容を認定された証としての資格取得がより求められる社会である。平成11年3月に新学習指導要領が発表され、普通高校においても教科「情報」が設けられ、情報教育が実施されるようになった。このことを受けて、本協会では新学習指導要領に沿い、かつ、「情報技術検定試験」にも対応できる問題集の発刊となった。

本標準問題集は、情報技術の進歩発展を考慮して最新情報を取り入れ、「情報技術基礎」の必要事項の確実な定着が図れるように例題と解き方を入れるなど工夫された内容になっている。より多くの方々が利用され、高校の情報に関する学習や資格取得に役立てば幸いである。

【改定の趣旨】

1. 新学習指導要領の「情報技術基礎」のねらいである情報技術の基礎的な知識と技術を習得できる内容にする。
2. 従来の演習問題集の内容を、時代の進展に合わせる。
3. 受検者が利用しやすく、簡潔で分かりやすい内容にする。

【構成と留意事項】

1. 「3級編 初級」 科目「情報技術基礎」学習者の初級程度を対象。
「2級編 中級」 情報技術学習者中級程度を対象。
「1級編 上級」 「基本情報技術者試験」の受験希望者を対象。
2. 内容には、情報技術の最新情報を取り入れる。
3. 理解を早め、基礎的事項の確実な定着を図るために、例題と解き方を入れる。
4. 過去に出題された本協会検定問題の活用を図る。
5. 卷末に、情報技術検定試験実施要項（抜粋）と内容一覧を掲載する。

平成23年4月

社団法人 全国工業高等学校長協会

1級情報技術検定試験問題集の編集には、次の編集委員があたりました。

(平成23年3月31日現在)

委 員 長 丸岡 俊之（大阪府立今宮工科高等学校長）

他 4名

編 集 事 務 橋本 三男（全国工業高等学校長協会事務局長）

《目 次》

【I ハードウェアの基礎知識】

1. 数の表現と処理

1. 1 情報量(ビット).....	1
1. 2 数の表現.....	2
1. 3 正の数・負の数.....	5
1. 4 固定小数点形式と浮動小数点形式.....	7
1. 5 データの表し方.....	9
1. 6 論理演算.....	11

2. コンピュータの基本回路

2. 1 基本論理回路と論理式.....	13
2. 2 演算回路(加算・一致・比較).....	17
2. 3 フリップフロップ(FF)を応用した回路.....	23
2. 4 エンコーダ・デコーダとマルチプレクサ・デマルチプレクサ.....	29
2. 5 インタフェース回路.....	31

3. コンピュータの基本構成と各部の働き

3. 1 基本構成.....	33
3. 2 中央処理装置.....	37
3. 3 主記憶装置.....	41
3. 4 助記憶装置.....	44
3. 5 入出力装置.....	48

4. 通信

4. 1 通信方式.....	52
4. 2 インターネット向けのブロードバンドサービス.....	57

5. ソフトウェアの基礎

5. 1 オペレーティングシステム.....	60
5. 2 プログラム言語.....	62
5. 3 アプリケーションソフトウェア.....	65
5. 4 ソフトウェアの開発.....	68

6. その他の情報関連知識

6. 1 情報技術と社会.....	70
6. 2 ネットワーク.....	71
6. 3 情報処理システム.....	75
6. 4 マルチメディア.....	78
6. 5 情報化の課題.....	80

《目次》

【II プログラミングの基礎知識】

7. アルゴリズム	【認定試験のエドノハナ】
7. 1 並べ換え（ソート）	既設システムの構成と機能
7. 2 探索	82
7. 3 順位付けと度数分布	85
7. 4 減化式	87
7. 5 簡単なアルゴリズム	89
7. 6 代表的なアルゴリズム	91
7. 7 定積分のアルゴリズム	94
7. 8 根を求めるアルゴリズム	100
	103
8. CASL II プログラムの作成能力	【認定・マー・質問】問題解説
8. 1 CASL II の基本的なプログラム	106
8. 2 CASL II の応用的なプログラム	120
9. C プログラムの作成能力	【認定・マー・質問】問題解説
9. 1 C の応用的なプログラム	136
9. 2 各種アルゴリズム	152
練習問題解答	166
アセンブラー言語の仕様	174
索引	180
付録	平成22年度の検定試験問題と解答

1. 数の表現と処理

1. 1 情報量（ビット）

要 点

コンピュータ内部の情報量の表現にはビット(bit)という単位が用いられている。ビット(bit)はbinary digitの略で、2進数1けたという意味をもち、2進数の0または1を決定できるような情報量を1ビットという。

- ・nビットでは、 2^n 個の情報を記憶できる。
- ・N個の情報を記憶するためには、 $\log_2 N$ ビット以上必要である。

□例題1.1□

「X」と「Y」の2種類の文字を1文字以上、最大n文字並べて符号を作るとき、100通りの符号を作るためのnの最小値を求めなさい。

解説

1文字のとき2通りの符号を作ることができる。同様に2文字のとき4通り、3文字のとき8通り、4文字のとき16通り、5文字のとき32通り、6文字のとき64通りで、1文字から6文字で合計126通りの符号を表現できる。したがって、nが6以上のとき100通りの符号を作ることができる。(X, Y, XX, XY, YX, YY, …)

解答 6

□例題1.2□

各バイトごとに番地が割り振られている 10^8 バイトの容量の記憶装置で、番地を識別するのに必要なビット数は最低何ビットか。ただし、 $\log_{10} 2 = 0.301$ として計算しなさい。また、 $\log_x Y = \frac{\log_2 Y}{\log_2 X}$ である。

解説

$$\log_2 10^8 = \frac{\log_{10} 10^8}{\log_{10} 2} = \frac{8}{0.301} \approx 26.6$$

解答 27ビット

練習問題

- 1 アルファベット26文字を2進符号化するために、最低何ビット必要か。
- 2 次の各間に答えなさい。
 - 6ビットで表現可能な符号は何個か。
 - 5枚の硬貨を1列に並べたとき表裏で表すことのできる状態は何個か。
 - ア～ンまでの仮名文字を、2進符号化するのに、最低何ビット必要か。
 - 英字のアルファベットA～Zと、数字の0～9の36文字を2進符号化するためには、最低何ビット必要か。
 - 英字のアルファベットA～Zとa～z、数字の0～9を2進符号化するために最低何ビット必要であるか。
- 3 12けたの16進数の最大値を10進数で表すと何けたになるか。ただし、 $\log_{10} 2 = 0.301$ とする。
- 4 0でないある10進数のけた数をDけたとするとき、これを2進数で表示したときのけた数Bとの関係を表す式はどれか。
ア.D ≈ 2 $\log_{10} B$ イ.D ≈ 10 $\log_2 B$ ウ.D ≈ B $\log_2 10$ エ.D ≈ B $\log_{10} 2$

1.2 数の表現

要 点

一般にnけたの正の整数Nは、rをradix(基數), dをdigit(ディジット)として,

$$\cdot N = d_n \times r^{n-1} + d_{n-1} \times r^{n-2} + \cdots + d_1 \times r^0$$
 と表される。

またnけたの小数以下の数Mは,

$$\cdot M = d_{-1} \times r^{-1} + d_{-2} \times r^{-2} + \cdots + d_{-n} \times r^{-n}$$
 と表される。

□例題1.3□

16進数の1AC.8について次の各間に答えなさい。

- ①10進数に変換しなさい。
- ②2進数に変換しなさい。
- ③8倍したものを16進数で表しなさい。

解説

①16進数の小数についても、10進数や2進数の小数と同じ考え方となる。各けたにそれぞれ16を基數とする重みがかかっている。

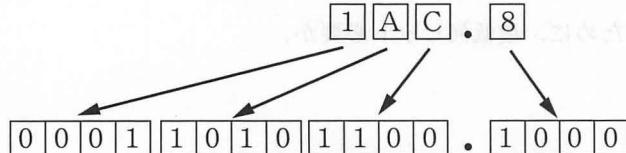
例えば $(1AC.8)_{16}$ を10進数に変換するときは下のように考える。

$$\begin{array}{ccccccccc} \dots & \boxed{\square} & \boxed{1} & \boxed{A} & \boxed{C} & \cdot & \boxed{8} & \boxed{\square} & \dots \\ \times & \times & \times & \times & \times & & \times & \times & \\ \dots & 16^3 & 16^2 & 16^1 & 16^0 & & 16^{-1} & 16^{-2} & \dots \end{array}$$

$$\begin{aligned} (1AC.8)_{16} &= (1)_{16} \times 16^2 + (A)_{16} \times 16^1 + (C)_{16} \times 16^0 + (8)_{16} \times 16^{-1} \\ &= 1 \times 16^2 + 10 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1} \end{aligned}$$

$$\begin{aligned} &= 1 \times 256 + 10 \times 16 + 12 \times 1 + 8 \times \frac{1}{16} \\ &= 256 + 160 + 12 + 0.5 \\ &= 428.5 \end{aligned}$$

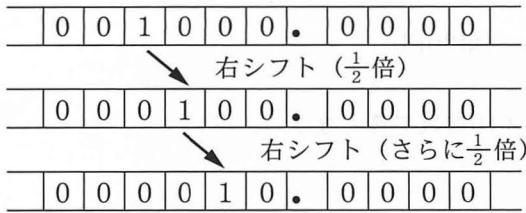
②図のように、2進数との関係は、16進数の1けたと2進数の4けたが対応するので、 $(1AC.8)_{16}$ を2進数に変換するには下のように考える。



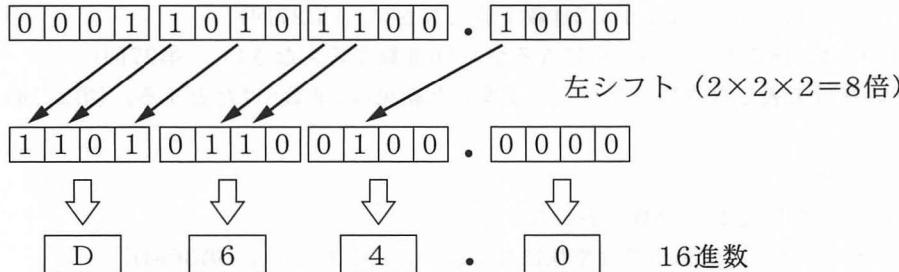
③16進数を2倍、4倍、8倍するときには、いったん2進数に変換しておくとわかりやすい。2進数を左に1けたシフトすると2倍、2けたシフトすると $2 \times 2 = 4$ 倍となる。

0	0	0	0	1	0	.	0	0	0	0
左シフト（2倍）										
0	0	0	1	0	0	.	0	0	0	0
左シフト（さらに2倍）										
0	0	1	0	0	0	.	0	0	0	0

同様に、右に1けたシフトすると $\frac{1}{2}$ 倍、2けたシフトすると $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ 倍となる。



したがって、 $(1AC.8)_{16}$ を8倍するには、 $(1AC.8)_{16} = (0001\ 1010\ 1100.1000)_2$ であるので、左に3けたシフトすればよい。



解答

① $(1AC.8)_{16} = (428.5)_{10}$ ② $(1AC.8)_{16} = (1\ 1010\ 1100.1)_2$ ③ $(D64)_{16}$

□例題1.4□

$113 - 64 = 38$ が成立するとき、何進法で計算しているか。

解説

与式がN進法であるとして、各けたの重みをつけた式を立ててNを求める。

$$(113)_N - (64)_N = (38)_N$$

$$(1 \times N^2 + 1 \times N + 3) - (6 \times N + 4) = (3 \times N + 8)$$

$$N^2 - 8N - 9 = 0$$

$$(N+1)(N-9) = 0$$

$$N > 0 \text{ より, } N = 9$$

解答 9進法

練習問題

5 A, Bがそれぞれ次の値のとき、各計算をおこない間に答えなさい。

(1) A, Bがそれぞれ $(0011\ 0010)_2$, $(0000\ 0101)_2$ である。(第36回)

$$(A \times B) \times (0.5)_{10} + (A \div B) \div (0.25)_{10}$$

①2進数で答えなさい。 ②16進数で答えなさい。

(2) A, Bがそれぞれ $(0011\ 0100)_2$, $(0001\ 0101)_2$ である。(第35回)

$$(A + B) \div (0.5)_{10} + (A - B) \div (0.25)_{10}$$

①2進数で答えなさい。 ②16進数で答えなさい。

(3) A, Bがそれぞれ $(1111\ 1010)_2$, $(0001\ 1001)_2$ である。(第34回)

$$(A \div B) \div (0.125)_{10}$$

①2進数で答えなさい。 ②16進数で答えなさい。

(4) A, Bがそれぞれ $(1010\ 1101)_2$, $(0000\ 1000)_2$ である。(第33回)

$$(A \div B) \div (0.5)_{10}$$

①2進数で答えなさい。 ②16進数で答えなさい。

(5) A, Bがそれぞれ $(1101\ 0111)_2$, $(0001\ 0000)_2$ である。(第32回)

$$(A \div B) \times (0.25)_{10}$$

①2進数で答えなさい。 ②16進数で答えなさい。

(6) A, Bがそれぞれ $(0110\ 0100)_2$, $(0100\ 1100)_2$ である。(第31回)

$$(A \times B) \div (64)_{10}$$

- ①2進数で答えなさい。 ②16進数で答えなさい。
- (7) A, Bがそれぞれ $(01100100)_2$, $(01001100)_2$ である。(第30回)
- $$(A - B) \times (2)_{10} + (A + B) \div (4)_{10}$$
- ①2進数で答えなさい。 ②16進数で答えなさい。 ③10進数で答えなさい。

6 次の各間に答えなさい。

- ①16進数のF3.1を2進数に変換しなさい。(第27回)
②16進数の6C.4を10進数に変換しなさい。(第27回)
③10進数の計算 37×16 はいくらになるか。2進数で答えなさい。(第27回)
④16進数1AB4を10進数の8で割るといいくらになるか。16進数で答えなさい。(第27回)
⑤10進数-19を2の補数で表しなさい。ただし、変換した結果は2進数8けたとする。(第27回)

7 次のそれぞれの間に答えなさい。

- (1) 16進数の小数 $(0.CA)_{16}$ を10進数の分数で表しなさい。

ただし、分数は約分して分母が最も小さな整数になるように表しなさい。(第36回)

$$(0.CA)_{16} = (\textcircled{1})_{10}$$

- (2) 16進数の小数 $(0.A4)_{16}$ を10進数の分数で表しなさい。

ただし、分数は約分して分母が最も小さな整数になるように表しなさい。(第35回)

$$(0.A4)_{16} = (\textcircled{2})_{10}$$

- (3) 16進数の小数 $(0.B8)_{16}$ を10進数の分数で表しなさい。

ただし、分数は約分して分母が最も小さな整数になるように表しなさい。(第34回)

$$(0.B8)_{16} = (\textcircled{3})_{10}$$

- (4) 次の10進数の計算結果を16進数で表しなさい。(第33回)

$$44.375 \div 4 = (\textcircled{4})_{16}$$

- (5) 次の計算結果を16進数で表しなさい。(第32回)

$$54.625 \times 4 = (\textcircled{5})_{16}$$

- (6) 次の計算結果を16進数で表しなさい。(第31回)

$$137 \div 64 = (\textcircled{6})_{16}$$

- (7) 次の計算結果を2進数で表しなさい。(第30回)

$$19 \div 32 = (\textcircled{7})_2$$

- (8) 次の計算結果を2進数で表しなさい。(第29回)

$$185 \div 256 = (\textcircled{8})_2$$

- (9) 次の計算結果を2進数で表しなさい。(第28回)

$$7 \div 16 = (\textcircled{9})_2$$

- (10) 次の計算結果を2進数で表しなさい。

$$9 \div 64 = (\textcircled{10})_2$$

8 16進数のA1F.4について次の各間に答えなさい。(第29回)

- ①10進数に変換しなさい。
②2進数に変換しなさい。
③ $\frac{1}{4}$ 倍したもの16進数で表しなさい。

9 $514 \div 5 = 102$ (あまり0)が成立するとき、何進法で計算しているか。

10 式 $(0.0101)_2 \div (0.001)_2$ を計算して、2進数で答えなさい。

11 次の10進数のうち、2進数で表すと無限小数になるものはどれか。

0.625 0.5 0.05 0.375

12 次の16進数の小数を分数で表しなさい。

- ①0.1 ②0.B ③0.01 ④0.FF ⑤0.A8

13 次の計算結果を2進数で表しなさい。

- (1) $1 \div 16 = (\textcircled{1})_2$ (2) $255 \div 256 = (\textcircled{2})_2$
(3) $29 \div 32 = (\textcircled{3})_2$ (4) $137 \div 64 = (\textcircled{4})_2$

14 次の各間に答えなさい。

- ①16進数の1C6.8を10進数に変換しなさい。
②16進数(4B)₁₆を8倍するとどうなるか。16進数で答えなさい。
③16進数13C8を10進数の8で割るといいくらになるか。16進数で答えなさい。
④16進数の3E5.2を10進数に変換しなさい。
⑤16進数1AB4に10進数の8をかけるといいくらになるか。16進数で答えなさい。
⑥16進数のE7.8を2進数に変換しなさい。

15 次の各間に答えなさい。

- ①16進数のA1.4を10進数に変換しなさい。
②16進数の計算 $47 - 3A$ の答えを求め、8進数で答えなさい。
③16進数の2B6を10進数に変換しなさい。
④10進数の365を2進化10進符号(BCDコード)で表しなさい。
⑤16進数の2F5.2を10進数に変換しなさい。
⑥10進数の753を2進化10進符号(BCDコード)で表しなさい。

16 次の10進数を2進数に変換しなさい。ただし、けた数は小数点以下8けたまでとし、それで表せる最も近い値で答えなさい。

- ①0.1 ②0.5 ③0.7 ④0.64 ⑤0.86

17 次の変換をしなさい。

- (1) $(10101100)_2 = (\textcircled{1})_8$
(2) $(A35D)_{16} = (\textcircled{2})_2$
(3) $(7326)_8 = (\textcircled{3})_2$
(4) $(C2B6)_{16} = (\textcircled{4})_8$
(5) $(11000101)_2 = (\textcircled{5})_{16}$
(6) $(5742)_8 = (\textcircled{6})_{16}$

1. 3 正の数・負の数

要 点

1語16ビットで正の整数を考えると、表現できる数は $0 \sim 65535 (0 \sim 2^{16}-1)$ となる。また最上位ビットが0のとき正の数、1のとき負の数とする2の補数表現で表すと、 $-32768 \sim 32767 (-2^{15} \sim 2^{15}-1)$ となる。

コンピュータで負の数を扱うときは2の補数を用いることが多い。

一般にr進数nけたの数Nの

- r の補数は $r^n - N$
- $r-1$ の補数は $r^n - N - 1$

で表される。

□例題1.5□

10進数76に対する2の補数を求めなさい。ただし2進数16けたとする。

解説

2の補数は $2^{16} - 76$ で求めるが、1の補数(ピット反転)を求めてから、1をくわえれば求まる。

$$\begin{array}{r}
 \text{10進数} & \text{2進数} \\
 \downarrow & \downarrow \\
 76 \cdots \cdots \cdots 0000\ 0000\ 0100\ 1100 \\
 & 1111\ 1111\ 1011\ 0011 \cdots \cdots \text{(1の補数)} \\
 +) & \underline{\hspace{1cm}} \quad 1 \\
 2^{16} - 76 \cdots \cdots 1111\ 1111\ 1011\ 0100 \cdots \cdots \text{(2の補数)}
 \end{array}$$

解答 1111 1111 1011 0100

□例題1.6□

次の計算を補数を用いて行いなさい。

① $98 - 76$ (10進数)

② $0111\ 1001 - 0100\ 0110$ (2進数)

解説

引く数の補数を求めて加算し、最上位へのけた上がりを無視する。

①引く数76の、10の補数は24

$$98 + 24 = \underline{122}$$

↑

最上位へのけた上がりを無視する。

②引く数 0100 0110 の2の補数は1011 1010

$$0111\ 1001 + 1011\ 1010 = \underline{1}\ 0011\ 0011$$

↑
最上位へのけた上がりを無視する。

解答 ①22 ②0011 0011

練習問題

18 次のそれぞれの間に答えなさい。

(1) 次の16ビットデータについて、文中の空欄を埋めなさい。(第36回)

$$1111\ 0110\ 1110\ 1010$$

このデータが2の補数を用いて表現されているとすれば、10進数では となる。

(2) 10進数の-72を、8ビットの2の補数で表しなさい。(第35回)

$$-72 = (\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_2$$

(3) 10進数の-30を2の補数で表しなさい。ただし、2進数8けたとする。(第34回)

$$-30 = (\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_2$$

(4) 10進数の-12を2の補数で表しなさい。ただし、2進数8けたとする。(第33回)

$$-12 = (\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_2$$

19 次の16ビットデータについて、文中の を埋めなさい。(第28回)

$$1111\ 0110\ 1100\ 1011$$

このデータを符号付きの2進数とすれば、10進数では である。

ただし、負の数の場合は2の補数で表現しているものとする。

20 2の補数で負の数を表したとき、16ビットで表現できる負の数で絶対値が最大である数値を16進数として表しなさい。

21 次の説明に最も関連しているものを解答群から選びなさい。

- ①多くの数値を加算するときに生じる誤差を少なくするために絶対値の昇順に加算する。
- ②ほぼ同じ大きさの数値の減算により有効数字のけた数が減少する。
- ③コンピュータで表現できる最大絶対値より大きい演算結果となった。
- ④コンピュータで表現できる最小絶対値より小さい演算結果となった。
- ⑤指定されたけた数におさまらない小さい部分に対して切り捨てを行った。
- ⑥有限なけた数の10進数の小数を、2進数の小数で表すと循環小数となった。

解答群

打切り誤差	アンダフロー	オーバフロー	丸め誤差	けた落ち	情報落ち
-------	--------	--------	------	------	------

22 次の各間に答えなさい。

- ①10進数の86に対する2の補数を求めなさい。ただし、2進数8けたとする。
- ② $(1010\ 1110)_2$ は、2の補数で表した負の値である。符号付きの10進数で表すといくらになるか。
- ③10進数76に対する2の補数を求めなさい。ただし、2進数16けたとする。
- ④10進数-13を2の補数で表しなさい。ただし、変換した結果は2進数8けたとする。

23 次の計算を2進数に変換して補数計算を行い、8進数で答えなさい。

- ① $37 - 28$ (10進数)
- ② $4A - 37$ (16進数)

24 次の10進数を2の補数(2進数8けた)で表し、結果を符号なしの論理データと考え16進数で答えなさい。

- ①-35
- ②-110
- ③-1
- ④-128

1.4 固定小数点形式と浮動小数点形式

要 点

一般に、固定小数点形式の数値の表し方においては、小数点の位置を最下位のけたの後に固定されているものとし表示する場合が多い。また、浮動小数点形式は、仮数と指数によって表現する方法で、必要なビットは表現する数によって異なる。

□例題1.7□

整数の16ビット固定小数点形式のデータ表現において、小数点の位置が最下位のけたの後に固定されているものとして、次の各間に答えなさい。ただし、負の数は2の補数で表現する。

- ① $(0111\ 1111\ 1111\ 1111)_2$ を10進数で答えなさい。(正の数の最大値)
- ② $(0000\ 0000\ 0000\ 0000)_2$ を10進数で答えなさい。
- ③ $(1111\ 1111\ 1111\ 1111)_2$ を10進数で答えなさい。
- ④ $(1000\ 0000\ 0000\ 0000)_2$ を10進数で答えなさい。(負の数の最小値)

解説

- ①

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

符号ビットが0、他のビットはすべて1

$$(2^{14} + 2^{13} + \dots + 2^1 + 2^0)_{10} = (2^{15} - 1)_{10} = (32767)_{10}$$

- ②

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

符号ビットが0、他のビットもすべて0

$$(0)_{10}$$

- ③

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

符号ビットが1、他のビットもすべて1

$$(-1)_{10}$$

④

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

符号ビットが1, 他のビットはすべて0

$$-(2^{15})_{10} = (-32768)_{10}$$

解答 ① $(32767)_{10}$ ② $(0)_{10}$ ③ $(-1)_{10}$ ④ $(-32768)_{10}$

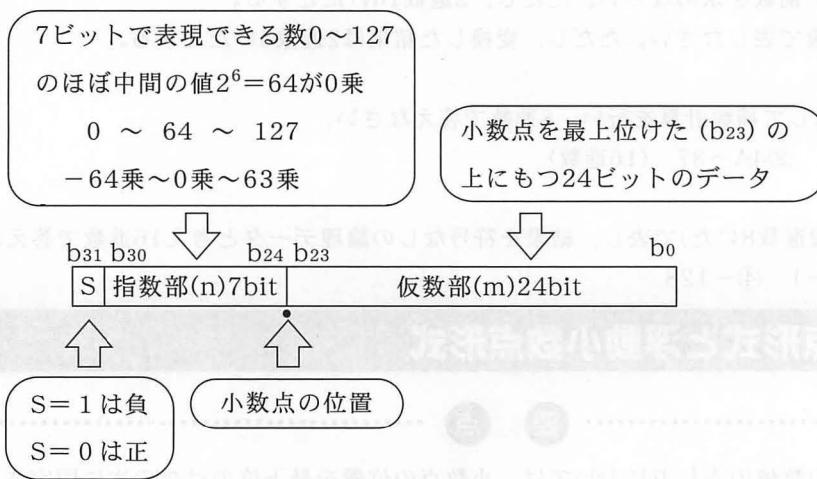
□例題1.8□

次の10進数の数値を、32ビット浮動小数点形式のデータ表現で表しなさい。ただし、基底を16として数値を、 $(0.\text{XXXXXX})_{16} \times 16^n$ で表したとき、符号部は正の数を0、負の数を1として1ビット、指数部は 16^n のnの値を $(n+64)_{10}$ として2進数7ビット、仮数部は $(0.\text{XXXXXX})_{16}$ (ただし、最上位のXは0以外)として2進数24ビットで表す。答えはビットパターンで答えなさい。

①17 ②0.046875

解説

①16を基底とする32ビット浮動小数点形式のデータ表現は図のようになる。



$$(17)_{10} = (11)_{16} = (0.11)_{16} \times 16^2 \text{ と表現できる。}$$

- ・符号部は1ビットで正 0
- ・指数部は7ビットで2 (データは64+2) 100 0010
- ・仮数部は24ビット0.11 .0001 0001 0000 0000 0000 0000

となり、 0100 0010 0001 0001 0000 0000 0000 0000 のビットパターンとなる。

$$\textcircled{2} 0.046875 \times 16 = 0.75, 0.75 \times 16 = 12 = (C)_{16}$$

$$\text{したがって } (0.046875)_{10} = (0.0C)_{16} = (0.C)_{16} \times 16^{-1}$$

- ・符号部は1ビットで正 0
- ・指数部は7ビットで-1(データは64-1), 011 1111
- ・仮数部は24ビット0.C .1100 0000 0000 0000 0000 0000

となり、 0011 1111 1100 0000 0000 0000 0000 0000 のビットパターンとなる。

解答

①0100 0010 0001 0001 0000 0000 0000 0000 ②0011 1111 1100 0000 0000 0000 0000 0000

練習問題

25 16ビット固定小数点形式のデータ表現において負の数を2の補数で表すとき、次の空欄に最も適切な数字を入れなさい。(第33回)

表現できる正の数の最大値は32767で、負の数の最小値は である。

26 次のそれぞれの間に答えなさい。

(1) 次の10進数を2進数に変換しなさい。ただし、小数部のけた数は8けた(小数第8位)までとし、それで表せる最も近い値で答えなさい。(小数第9位以下は切り捨てる)(第32回)

$$89.64 = (\boxed{①})_2$$

(2) 次の10進数を2進数に変換しなさい。ただし、小数部のけた数は8けた(小数第8位)までとし、それで表せる最も近い値で答えなさい。(小数第9位以下は切り捨てる)(第31回)

$$35.7 = (\boxed{②})_2$$

27 次の説明に最も適切な語を解答群から選びなさい。(第31回)

数値を **①** と **②** によって表す方法を浮動小数点表示という。例えば、10進数の浮動小数点の演算結果として、 0.000235×10^7 が得られたとき、**①** は、0.000235、**②** は 10^7 の7となる。これを標準の形式 0.235×10^4 に変えることを **③** という。

10進数の17を16進数で表して **③** すると、 $0.\boxed{④} \times 16^{\boxed{⑤}}$ になる。

解答群

ア. 実数	イ. 仮数	ウ. 標準化	エ. 指数	オ. 整数	カ. 小数点	キ. 正規化	
ク. 1	ケ. 2	コ. 3	サ. 4	シ. 5	ス. 6	セ. 7	ソ. 8
ツ. 11	テ. 12	ト. 13	ナ. 14	ニ. 15	ヌ. 16	ネ. 17	タ. 9 チ. 10

28 10進数0.375を、32ビット浮動小数点形式のデータ表現で表しなさい。ただし、基底を2として数値を $(0.1XX\cdots X)_2 \times 2^n$ で表したとき、符号部は正の数を0、負の数を1として1ビット、指数部は 2^n のnの値を2の補数表示で2進数7ビットで表し、仮数部は $(0.1XX\cdots X)_2$ として2進数24ビットで表す。答えはビットパターンで答えなさい。

29 16ビットのデータを左に2ビットシフトすると、あふれが生じ、得られた値は16進数でAF34となった。
元のデータを16進数で表しなさい。

30 次の16ビットデータについて各間に答えなさい。

1011 1100 1100 1010

①16進数で表すといらか。

②符号なし2進数とするといらか。10進数で答えなさい。

③符号付きの2進数とすればといらか。10進数で答えなさい。ただし、負の数は2の補数で表現する。

④小数点がデータのビット3とビット4の間にあるとするといらか。ただし符号なしとし、10進数で答えなさい。(最下位ビットをビット0とする)

⑤符号部をビット15(最上位ビット)、指数部をビット8~14、仮数部をビット0~7とすると16進数でいくらか。ただし、基底を16として数値を $(0.XX)_{16} \times 16^n$ で表したとき、符号部は正の数を0、負の数を1として1ビット、指数部は 16^n のnの値を $(n+64)_{10}$ として、仮数部は $(0.XX)_{16}$ (ただし、最上位のXは0以外) として表す。

1. 5 データの表し方

要 点

コンピュータで扱うデータは、数値のほかに文字や記号がある。それを表現するのに用いる符号がコードである。ISO、ASCII、JISコード等がある。

数字や英文字のように8ビットで表現されるコードを1バイトコードといい、最大 $2^8 = 256$ 個のデータを表現できる。漢字のように16ビットで表現されるコードを2バイトコードといい、最大 $2^{16} = 65536$ 個のデータを表現できる。

□例題1.9□

文字を表す8ビットのコードのうち、最上位ビットを偶数パリティビット、他の7ビットを文字コードとするとき、16進数で表現された7ビットの文字コード41, 53, 7Aに偶数パリティビットを付加して16進数で表しなさい。

解説

偶数パリティでは、2進数で表されたビットパターンの1の数が偶数になるようにパリティビットを付加する。

$$(41)_{16} = (100\ 0001)_2 \quad 1\text{の個数}2\text{個} \rightarrow \text{偶数パリティビット}0 \rightarrow (0100\ 0001)_2 = (41)_{16}$$

$$(53)_{16} = (101\ 0011)_2 \quad 1\text{の個数}4\text{個} \rightarrow \text{偶数パリティビット}0 \rightarrow (0101\ 0011)_2 = (53)_{16}$$

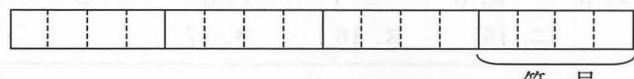
$$(7A)_{16} = (111\ 1010)_2 \quad 1\text{の個数}5\text{個} \rightarrow \text{偶数パリティビット}1 \rightarrow (1111\ 1010)_2 = (FA)_{16}$$

解答 41, 53, FA

練習問題

31 10進数を次のきまりにしたがって表現するとき、文中の空欄を埋めなさい。(第36回)

パック形式では、10進数の各けたをBCDと同じ方法で4ビットの2進数で表す。ただし、最下位の4ビットに、正の数の場合1100、負の数の場合1101の符号をつけてわえる。



このとき、-834は、□と表現できる。

32 次のきまりにしたがって表現するとき、文中の空欄を埋めなさい。

コンピュータで文字データを以下のように扱うとする。1バイトの中をゾーンビットとして4ビット、データビットとして4ビットに分けて用い、英字のA～Iはゾーンビットが1100、J～Rは1101、S～Zは1110とする。また数値の場合はゾーンビットが1111とする。データビットについては、それぞれのゾーンごとの昇順に並べた文字データを0001から順に当てはめる。この約束にしたがって次の文字を1バイトで表現しなさい。ただし、ビット0を最下位ビット、ビット7を最上位ビットとしビット0から3をデータビット、ビット4から7をゾーンビットとする。

このとき、文字データ F は、□で表される。

33 次の説明に最も適した語句を解答群より選びなさい。

- ①拡張2進化10進符号、IBM社が自社の汎用コンピュータ用として規格したコードで、汎用コンピュータの標準規格。
- ②ASCIIコードをもとにISO(国際標準化機構)が国際規格として制定。
- ③ANSI(アメリカ規格協会)が制定。7ビットの符号と1ビットのパリティービットで1文字を表す。
- ④ISOコードに片仮名や漢字を加えたもので、漢字1文字を2バイト(16ビット)で表す。
- ⑤パソコンで使用されている文字コードで、1バイトと2バイトの文字コードを混在させたもの。JISコードと漢字コード部分の互換性はない。
- ⑥拡張UNIXコード、世界各国の異なった文字に対応できるコード。
- ⑦ISO/IEC(国際電気標準会議)が制定。世界各国の異なった文字に対応させるため、すべての文字を2バイトで表す。

解答群

- | | | | |
|-----------------------|------------|-------------------|-----------|
| ア. シフトJISコード | イ. Unicode | ウ. ASCII(アスキー)コード | エ. ISOコード |
| オ. EBCDIC(エビシディック)コード | カ. JISコード | キ. EUCコード | |

34 コンピュータで文字データを扱う場合、1バイトの中をゾーンビットとして4ビット、データビットとして4ビットに分けて用いる場合がある、英字のA～Iはゾーンビットが1100、J～Rは1101、S～Zは1110である。

また数値の場合はゾーンビットが1111となる。この約束に従って次の文字を1バイトで表現しなさい。ただし、ビット4から7をゾーンビット、ビット0から3をデータビットとして、A, J, Sのデータビットは0001であるとする。

- ①B ②I ③M ④9

35 10進数の-1345をパック10進数(パック形式)で表したときのビットパターンをいいなさい。ただし、符号は最下位の4ビットとし、正の数のとき1100、負の数のとき1101とする。

36 10進数の+463をゾーン10進数(アンパック形式)で表したときのビットパターンをいいなさい。ただし、符号は4ビットとし、正の数のとき1100、負の数のとき1101とする。ゾーン10進数では、10進数の各けたを8ビットで表し、それぞれのけたの上位4ビットをゾーンビットといい数値を表す0011を入れ、10進数の最下位けたのゾーンビットに符号を入れるものとする。

37 JIS8単位符号は、下位7ビットが文字を表現する(区別する)情報で、最上位ビットはパリティビットといわれ、誤り検出用のビットである。いま偶数パリティ方式として、パリティビットを附加した場合、次にあげる文字の符号を16進数で答えなさい。ただし文字Aのパリティビットなしの16進数は41で、文字0は30である。

- ①D ②F ③U ④3 ⑤8

1. 6 論理演算

要 点

論理積や論理和などをもつて、2進数のデータをビットごとに演算することを論理演算という。

X = A · B		
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

論理積
AND

X = A+B		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

論理和
OR

X = \bar{A}	
A	X
0	1
1	0

否定
NOT

X = A \oplus B		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

排他的論理和
EX-OR

X = $\bar{A} \cdot B$		
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

否定論理積
NAND

X = $\bar{A} + B$		
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

否定論理和
NOR

(排他的論理和exclusive-ORはEX-OR、 XOR、 EORなどに略される)

□例題1.10□

ビット列A(1000 1010)₂とビット列B(1111 1111)₂の排他的論理和の演算を行いなさい。

解説

各ビットごとに排他的論理和の演算をおこなう。

$$\begin{array}{r} \text{ビット列A} \quad 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ \text{EX-OR} \quad \text{ビット列B} \quad 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \hline 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \end{array}$$

この演算では、ビット列Aをビット列Bを用いて反転させていることがわかる。

解答 (0111 0101)₂

練習問題

38 次の説明に最も関連しているものを解答群から選びなさい。

- ①8ビットのデータの下位3ビットはそのままで、上位5ビットを反転させる。
- ②8ビットのデータの下位3ビットはすべて0にして、上位5ビットのみを取り出す。
- ③8ビットのデータの下位3ビットはそのまま、上位5ビットをすべて1にする。
- ④8ビットのデータの下位3ビットはすべて1にして上位5ビットをすべて反転させる。

解答群

16進数F8と論理積をとる。

16進数F8と排他的論理和をとる。

16進数F8と否定論理和をとる。

16進数F8と論理和をとる。

16進数F8と否定論理積をとる。

- 39 2の補数で負の数を表したとき、8ビットの2進数nに対し $-n$ を求める式はどれか。ただし、+は加算、ORはビットごとの論理和、XORはビットごとの排他的論理和を表す。

解答群

- | | |
|---|---|
| ア. $(n \text{ OR } 10000000) + 00000001$ | イ. $(n \text{ OR } 11111110) + 11111111$ |
| ウ. $(n \text{ XOR } 10000000) + 11111111$ | エ. $(n \text{ XOR } 11111111) + 00000001$ |

- 40 8ビットのレジスタの各ビットの値をP, D6, D5, D4, D3, D2, D1, D0とし、D0～D6は7ビットの文字データ、Pは奇数パリティビットであるとき、成立する関係式はどれか。ただし、⊕は排他的論理和を表す。

解答群

- | |
|---|
| ア. $P \oplus D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0 = 0$ |
| イ. $P \oplus D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0 = 1$ |
| ウ. $0 \oplus D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0 = P$ |
| エ. $D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0 = P$ |

- 41 32ビットの浮動小数点を次の形式で表現するとき、32ビットのうち、指数部を取り出すためにビットごとのAND演算を行う際に使うビットパターンを16進数で表しなさい。

S	E(8ビット)	M(23ビット)
---	---------	----------

S : 符号部(1ビット) E : 指数部(8ビット) M : 仮数部(23ビット) (IEEE 754 (1985) 標準)

2. コンピュータの基本回路

2. 1 基本論理回路と論理式

要 点

論理回路において電圧のH(高い), L(低い)は、論理数学の1, 0に対応しており、回路と論理式の間には密接な関係がある。

[1] 論理代数(プール代数)

論理代数は0か1の2値状態のみを扱い、コンピュータの論理の基本となる論理数学である。

<定理>

恒等	$A \cdot 0 = 0$	$A + 1 = 1$	$A \cdot 1 = A$	$A + 0 = A$
同一	$A \cdot A = A$	$A + A = A$		
補完	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$		
復元	$\bar{\bar{A}} = A$			
交換	$A \cdot B = B \cdot A$	$A + B = B + A$		
結合	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$		
分配	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + B \cdot C = (A + B) \cdot (A + C)$		
吸収	$A + A \cdot B = A$			
ド・モルガン	$\overline{A+B+C+\dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$	$\overline{A \cdot B \cdot C \cdot \dots} = \bar{A} + \bar{B} + \bar{C} + \dots$		

以上の定理は、 \cdot と $+$, 1と0を同時に入れ替ても成立する、(上記の左右の関係)

[2] 真理値表

真理値表は入出力の関係を図表に表し、論理式やカルノー図を作ったりする際に使われる。

例として全加算回路の真理値表を示す。(Sn : 加算結果 Cn : けた上り)

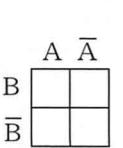
入力論理	入力			出力	
	A	B	C	Sn	Cn
$\bar{A} \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0
$\bar{A} \cdot \bar{B} \cdot C$	0	0	1	1	0
$\bar{A} \cdot B \cdot \bar{C}$	0	1	0	1	0
$A \cdot \bar{B} \cdot C$	0	1	1	0	1
$A \cdot \bar{B} \cdot \bar{C}$	1	0	0	1	0
$A \cdot B \cdot \bar{C}$	1	0	1	0	1
$A \cdot B \cdot C$	1	1	0	0	1
$A \cdot B \cdot C$	1	1	1	1	1

ここで真理値表を論理式で表すと、

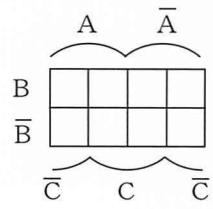
$$\begin{aligned}
 Sn &= \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C \\
 &= (\bar{A} \cdot \bar{B} + A \cdot B) \cdot C + (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \bar{C} \\
 &= (\bar{A} \oplus B) \cdot C + (A \oplus B) \cdot \bar{C} = (A \oplus B) \oplus C \\
 Cn &= \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C \\
 &= (\bar{A} \cdot B + A \cdot \bar{B}) \cdot C + A \cdot B \cdot (\bar{C} + C) \\
 &= (A \oplus B) \cdot C + A \cdot B
 \end{aligned}$$

[3] カルノー図

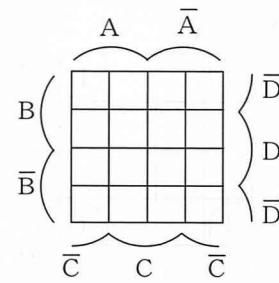
論理式を簡略化したり論理回路を設計する場合に、論理代数よりカルノー図はミスがなく簡単である。



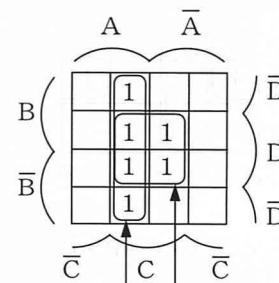
2変数



3変数



4変数



$$\begin{aligned}
 X &= A \cdot C + C \cdot D \\
 &= C \cdot (A + D)
 \end{aligned}$$

<利用方法>

論理式の各項に相当する区画を記号“1”で埋めて、全体を眺め、“1”を囲んでブロック化する。ブロック化は、縦または横に、あるいは縦横とも連続する2のn乗個の区画を囲み、できるだけ大きくブロック化する。ブロック相互は重なる部分があつてもよい。ブロック化した部分をすべて論理積の式に書き表しそれの論理和を作る。図表は地球儀を世界地図に広げたと同様に、それぞれ上下、左右につながっていると解釈する。

$$X = A \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot C \cdot D + A \cdot \bar{B} \cdot C + A \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot C \cdot D$$

この式をカルノー図で簡略化すると図のようになる。

□例題2.1□

次の各間に答えなさい。

- (1) A・B・C 3個のスイッチがある。このスイッチのうち少なくとも2個がONになったときシステムが動作するための論理式を求めなさい。

- (2) 次の論理式を簡略化し論理回路で表しなさい。

$$X_1 = A \cdot B \cdot \bar{C} + A \cdot C + \bar{A} \cdot B \cdot \bar{C}$$

$$X_2 = A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

解説

- (1) この真理値表は右のようになり論理式は、

$$X = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

① ② ③ ④

になる。

カルノー図と、ブール代数のいずれかを用いて、式を簡略化できる。

[カルノー図による簡略化]

論理式より、カルノー図を描く。

A	\bar{A}	B	1	1	1	
\bar{B}			1			
\bar{C}		C				\bar{C}

これより、次式のように

簡略化できる。

$$X = A \cdot B + A \cdot C + B \cdot C = A \cdot (B + C) + B \cdot C$$

[ブール代数による簡略化]

$$X = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

$$= \bar{A} \cdot B \cdot C + \underline{A \cdot B \cdot C} + A \cdot \bar{B} \cdot C + \underline{A \cdot B \cdot C} + A \cdot B \cdot \bar{C} + \underline{A \cdot B \cdot C}$$

$$= (\bar{A} + A) \cdot B \cdot C + A \cdot (B + \bar{B}) \cdot C + A \cdot B \cdot (\bar{C} + C)$$

$$= B \cdot C + A \cdot C + A \cdot B$$

$$= A \cdot (B + C) + B \cdot C$$

- (2) カルノー図と、ブール代数のいずれかを用いて、式を簡略化できる。

[カルノー図による簡略化]

論理式より、右のようにカルノー図を描く。

X ₁	A	\bar{A}	B	1	1	1	
			\bar{B}	1			
			\bar{C}		C		\bar{C}

$$X_1 = A \cdot C + B \cdot \bar{C}$$

X ₂	A	\bar{A}	B	1			\bar{D}
			\bar{B}	1			D
			\bar{C}		C		\bar{C}

$$X_2 = \bar{C} \cdot \bar{D}$$

[ブール代数による簡略化]

$$X_1 = A \cdot B \cdot \bar{C} + A \cdot C + \bar{A} \cdot B \cdot \bar{C}$$

$$= (A + \bar{A}) \cdot B \cdot \bar{C} + A \cdot C$$

$$= B \cdot \bar{C} + A \cdot C$$

$$X_2 = A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$= A \cdot \bar{C} \cdot \bar{D} \cdot (B + \bar{B}) + \bar{A} \cdot \bar{C} \cdot \bar{D} \cdot (B + \bar{B})$$

$$= A \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{C} \cdot \bar{D}$$

$$= (A + \bar{A}) \cdot \bar{C} \cdot \bar{D}$$

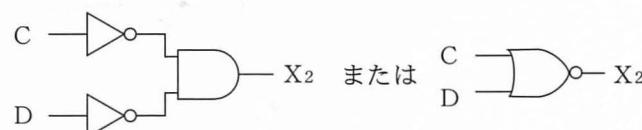
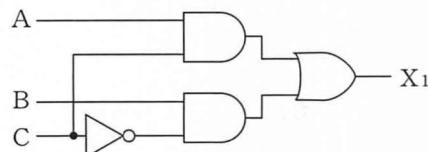
$$= \bar{C} \cdot \bar{D}$$

解答

$$(1) A \cdot (B + C) + B \cdot C$$

$$(2) X_1 = B \cdot \bar{C} + A \cdot C \quad X_2 = \bar{C} \cdot \bar{D}$$

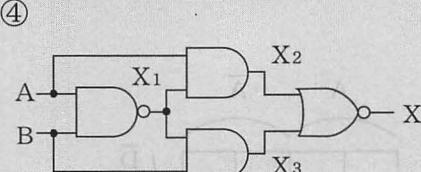
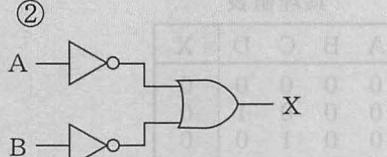
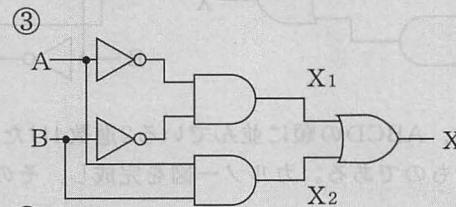
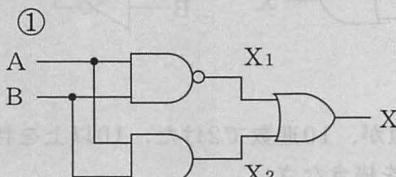
$$\text{または } X_2 = \bar{C} + D$$



(ド・モルガンの定理)

□例題2.2□

下図の各論理回路Xの出力の状態を表す論理式を解答群から選びなさい。



解答群

ア. $X = \bar{A} \cdot \bar{B}$

イ. $X = \bar{A} \cdot B$

ウ. $X = A \cdot \bar{B} + \bar{A} \cdot B$

エ. $X = A \cdot B + \bar{A} \cdot \bar{B}$

オ. $X = 1$

解説

① $X_1 = \bar{A} \cdot \bar{B} \quad X_2 = A \cdot B$

$$X = \bar{A} \cdot \bar{B} + A \cdot B = 1$$

② $X = \bar{A} + \bar{B}$ となるが答がないのでド・モルガンの定理を使う。

③ $X_1 = \bar{A} \cdot \bar{B} \quad X_2 = A \cdot B \quad X = \bar{A} \cdot \bar{B} + A \cdot B$

④ $X_1 = \bar{A} \cdot B \quad X_2 = A \cdot X_1 \quad X_3 = B \cdot X_1$

$$X = X_2 + X_3 = \overline{A \cdot (\bar{A} \cdot B) + B \cdot (\bar{A} \cdot B)}$$

$$= \overline{A \cdot (\bar{A} + B)} + B \cdot (\bar{A} + B)$$

$$= \overline{A \cdot \bar{A}} + A \cdot \bar{B} + B \cdot \bar{A} + B \cdot \bar{B}$$

$$= \overline{A \cdot \bar{B}} + B \cdot \bar{A} = (\overline{A \cdot \bar{B}}) \cdot (B \cdot \bar{A})$$

$$= (\overline{A} + \bar{\bar{B}}) \cdot (\bar{B} + \bar{A}) = (\overline{A} + B) \cdot (\bar{B} + A)$$

$$= \overline{A} \cdot \bar{B} + \bar{A} \cdot A + B \cdot \bar{B} + \bar{A} \cdot B$$

$$= \overline{A} \cdot \bar{B} + A \cdot B$$

解答

- ① オ ② イ ③ エ ④ エ

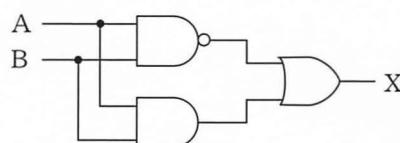
練習問題

- 1 次の論理式で表される論理回路を解答群から選びなさい。(第31回)

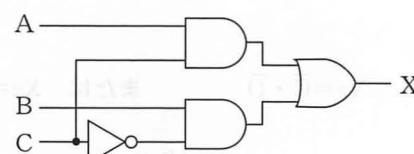
- ① $X = A \cdot B + \bar{A} \cdot \bar{B}$
- ② $X = A \cdot \bar{B} + \bar{A} \cdot B$
- ③ $X = 1$
- ④ $X = A \cdot B \cdot \bar{C} + A \cdot C + \bar{A} \cdot B \cdot \bar{C}$
- ⑤ $X = \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$

解答群

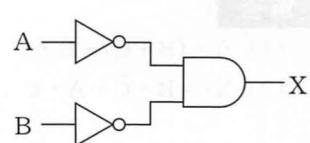
ア.



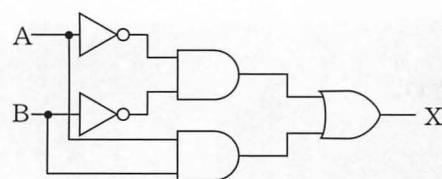
イ.



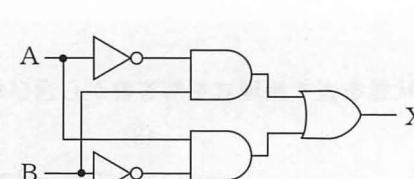
ウ.



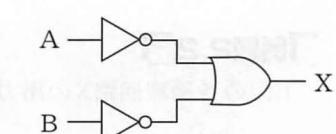
エ.



オ.

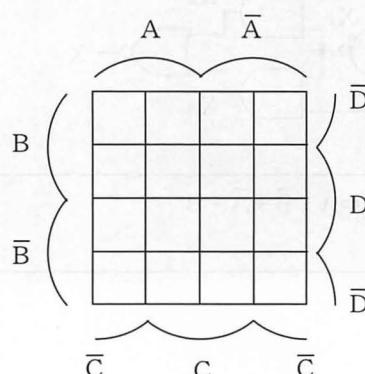


カ.



- 2 次の真理値表は、ABCDの順に並んでいる2進数4けたの数字の値が、10進数で2けた、10以上を判断する論理回路を示したものである。カルノー図を完成し、その論理回路を描きなさい。

真理値表



A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- 3 次の論理式を簡略化したものを解答群から選びなさい。

- ① $X_1 = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$
- ② $X_2 = A \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C$
- ③ $X_3 = A \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot C$
- ④ $X_4 = (\bar{A} + B + C) \cdot (A + B + \bar{C}) \cdot (A + B + C)$

解答群

ア. $A \cdot B + \bar{A} \cdot C$

イ. $\bar{A} \cdot \bar{B} \cdot \bar{C}$

ウ. $A + \bar{B} \cdot \bar{C}$

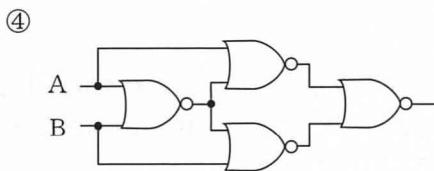
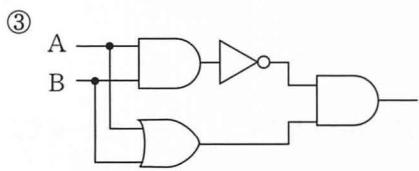
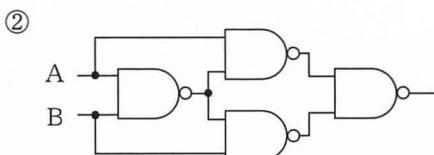
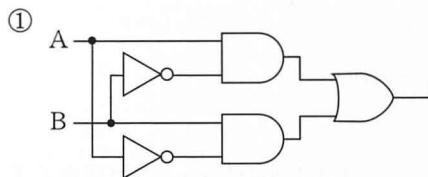
エ. $A \cdot C + B$

オ. $A \cdot C + \bar{B}$

カ. $\bar{A} \cdot \bar{B} + \bar{C}$

キ. $\bar{A} \cdot \bar{C} \cdot \bar{B}$

4 次の論理回路の出力に相当する論理式を解答群から選びなさい。答は重複して選んでもよい。



解答群

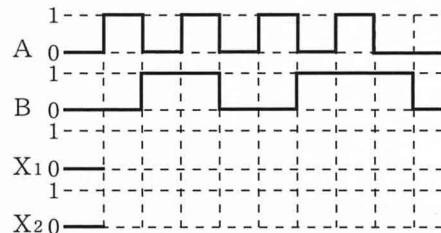
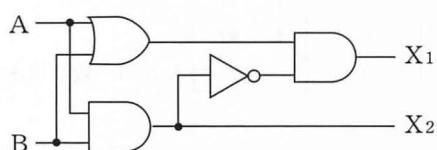
ア. $A \cdot B + \bar{A} \cdot \bar{B}$

イ. $\bar{A} \cdot B + A \cdot \bar{B}$

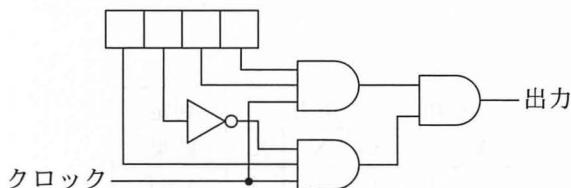
ウ. $A \cdot B + \bar{A} \cdot \bar{B}$

エ. $A \cdot B$

5 図のような論理回路の入力A, Bに右図で示すような入力波形が加えられたとき、出力波形を記入しなさい。



6 4ビットのレジスタと論理ゲートが図のように接続されている。クロック入力に“1”的パルスを加えたとき、出力が“1”になるのは、レジスタの内容がどのようなときか。

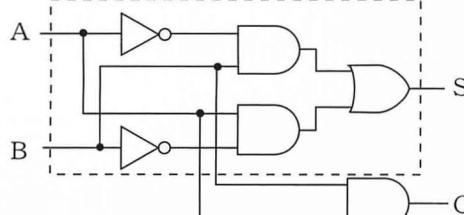


2. 2 演算回路（加算・一致・比較）

要 点

[1] 加算回路

①半加算回路(HA:ハーフアダマー)



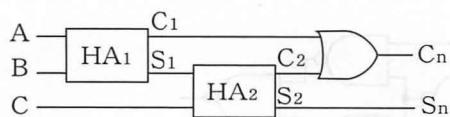
$$S = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$

$$C = A \cdot B$$



入力		出力	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

②全加算回路(FA : フルアダード)



$$\begin{aligned} S_n &= (A \oplus B) \cdot \bar{C} + (\bar{A} \oplus \bar{B}) \cdot C \\ &= (A \oplus B) \oplus C \\ C_n &= A \cdot B + (A \oplus B) \cdot C \end{aligned}$$

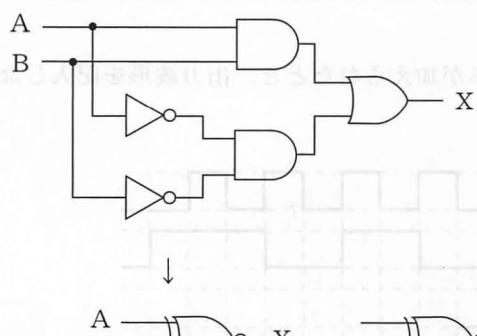
(データ : A, B, 加算結果 : S, けた上がり : C_n, Cは下から上がり)

[2] 一致回路

一致回路はAとBのデータが等しいかどうかを判定する。

真理値表より、論理式は、 $X = X = A \cdot B + \bar{A} \cdot \bar{B}$ となり、論理回路は図のようになる。したがって、EX-NOR回路に置き換えることもできる。

この回路を複数ビット分組み合わせ、すべてのビットが一致しているかどうか、すなわち、データが一致しているかどうか、比較できる。



入力		出力	判定 結果
A	B	X	
0	0	1	一致
0	1	0	不一致
1	0	0	不一致
1	1	1	一致

入力が一致したとき 1 出力

↓

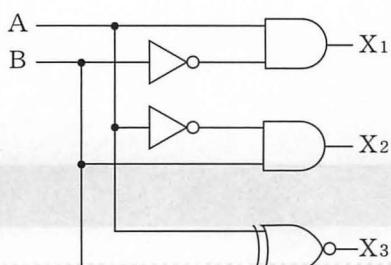
入力が一致したとき 1 出力

$$A \quad \text{---} \quad B \quad \text{---} \quad X = \overline{\overline{A} \overline{B}} = \overline{A} \overline{B}$$

EX-NOR回路

[3] 比較回路

比較回路はAとBのデータの大小関係を判定する。真理値表より、論理回路は図のようになる。

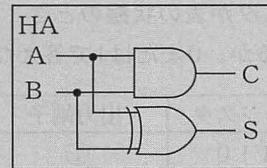
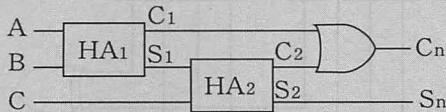


入力	A	B	出力			判定 結果
			X ₁	X ₂	X ₃	
0	0	0	0	0	1	A=B
0	1	0	0	1	0	A < B
1	0	1	1	0	0	A > B
1	1	1	0	0	1	A=B

被除数	除数	商	余り
0	0	0	0
0	1	0	0
1	1	1	0
1	0	0	1
1	1	1	1

□例題2.3□

次の図に関して各問に答えなさい。



①HAの内部はANDとEX-ORからなる回路であると考え、 HA₁の出力S₁, C₁の論理式を解答群から選びなさい。

解答群

- ア. $A \cdot B$ イ. $A+B$ ウ. $\bar{A} \cdot \bar{B} + A \cdot B$ エ. $\bar{A} \cdot B + A \cdot \bar{B}$

②HA₂の出力S₂, C₂の論理式を解答群から選びなさい。

解答群

- ア. $\bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$
 イ. $\bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C$
 ウ. $\bar{A} \cdot B \cdot C + A \cdot B \cdot \bar{C}$
 エ. $\bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C$

③S_n, C_nの状態を表す論理式の項の一つとして含まれるもの解答群から選びなさい。複数個あればその全部を答えなさい。

解答群

- ア. $A \cdot B \cdot C$ イ. $\bar{A} \cdot B \cdot \bar{C}$ ウ. $\bar{A} \cdot \bar{B} \cdot \bar{C}$ エ. $\bar{A} \cdot B \cdot C$
 オ. $A \cdot \bar{B} \cdot \bar{C}$ カ. $\bar{A} \cdot \bar{B} \cdot C$ キ. $A \cdot B \cdot \bar{C}$ ク. $A \cdot \bar{B} \cdot C$

解説

①HA₁は半加算器であるので、 $S_1 = A \oplus B = \bar{A} \cdot B + A \cdot \bar{B}$, $C_1 = A \cdot B$ である。

②HA₂は半加算器であるので、

S_2 の式に $S_1 = \bar{A} \cdot B + A \cdot \bar{B}$ を代入すると、

$$\begin{aligned} S_2 &= S_1 \oplus C = \bar{S}_1 \cdot C \\ &= (\bar{A} \cdot B + A \cdot \bar{B}) \cdot C \\ &= (\bar{A} \cdot B) \cdot (A \cdot \bar{B}) \cdot C \\ &= (\bar{A} + \bar{B}) \cdot (\bar{A} + \bar{B}) \cdot C \\ &= (A + \bar{B}) \cdot (\bar{A} + B) \cdot C \\ &= (A \cdot \bar{A} + A \cdot B + \bar{A} \cdot \bar{B} + \bar{B} \cdot B) \cdot C \\ &= (0 + A \cdot B + \bar{A} \cdot \bar{B} + 0) \cdot C \\ &= A \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C \end{aligned}$$

$$\begin{aligned} &\quad + S_1 \cdot \bar{C} \\ &\quad + (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \bar{C} \\ &\quad + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} \\ &\quad + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} \\ &\quad + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} \\ &\quad + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} \end{aligned}$$

C_2 の式に $S_1 = \bar{A} \cdot B + A \cdot \bar{B}$ を代入すると

$$\begin{aligned} C_2 &= S_1 \cdot C = (\bar{A} \cdot B + A \cdot \bar{B}) \cdot C \\ &= \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C \end{aligned}$$

③ $S_n = S_2 = A \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C}$

$$\begin{aligned} C_n &= C_1 + C_2 = A \cdot B + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C \\ &= A \cdot B \cdot (C + \bar{C}) + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C \\ &= A \cdot B \cdot C + A \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C \end{aligned}$$

解答 ① S_1 : エ , C_1 : ア

② S_2 イ , C_2 エ

③ S_n : ア イ オ カ , C_n : ア エ キ ク

□例題2.4□

図の論理回路について、次の各間に答えなさい。(第29回)

- (1) Xレジスタ、Yレジスタが表の状態のとき、出力端子の状態①、②はどうなるか。0または1で答えなさい。

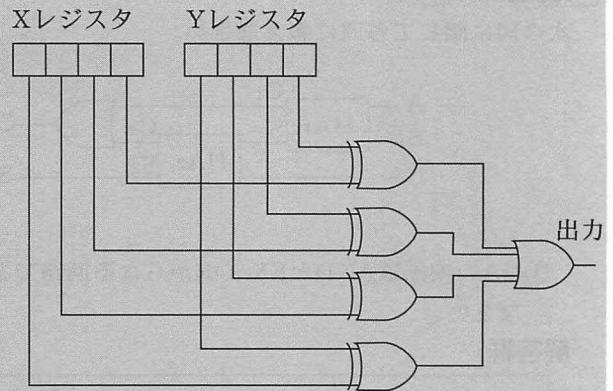
Xレジスタ	Yレジスタ	出力端子
1 0 1 1	0 1 1 0	①
1 0 0 1	1 0 0 1	②

- (2) この論理回路の説明について文中の空欄に入る最も適切な語を解答群から選びなさい。

Xレジスタ、Yレジスタのそれぞれの対応するビットがEX-OR回路で接続されており、各ビットごとの不一致を検出する。更にこれらの全ビットごとの出力がOR回路に入力されているため、OR回路のすべての入力(すべてのEX-OR回路の出力)が③のときのみ出力④、すなわち対応する全ビットが一致しているとき、④を出力する。すなわち、⑤回路であることがわかる。

解答群

- ア. 0 イ. 1 ウ. 一致検出 エ. パリティーチェック オ. キャリー検出 ハ. 比較



解説

Xレジスタ、Yレジスタのそれぞれの対応するビットがEX-OR回路で接続されており、各ビットごとに一致しているとき0(不一致のとき1)を出力する。さらにこれらの全ビットごとの出力がOR回路に入力されているため、OR回路の全入力(EX-OR回路の全出力)がすべて0のときのみ出力0、すなわち対応する全ビットが一致しているとき、0を出力する。

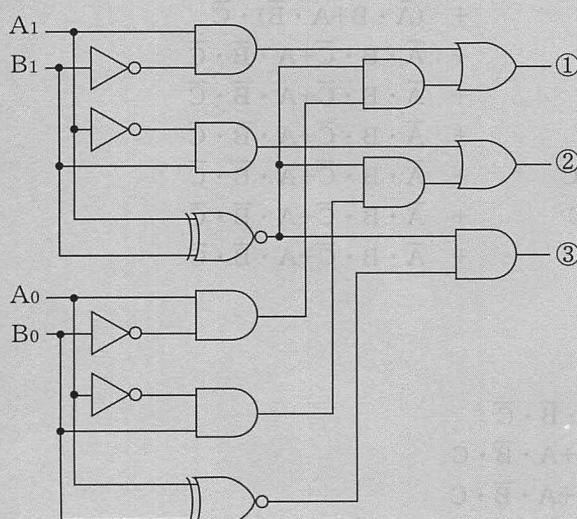
出力が0のとき対応する全ビットが一致していることを検出できる。

解答

- ①1 ②0 ③ア ④エ ⑤ウ

□例題2.5□

図の論理回路は2ビットのデータの比較回路である。①～③の判定結果を解答群から選びなさい。ただし、A₀～A₁はデータA、B₀～B₁はデータBの各ビットを表し、A₀、B₀が下位ビットである。



解答群

- ア. A < B
イ. A = B
ウ. A > B

解説

左側の論理回路は、A₁とB₁、A₀とB₀のそれぞれを1ビットずつ比較する回路である。それらの判定結果により、更に右側の論理回路で①、②は大小を、③はすべて一致を判定している。

解答

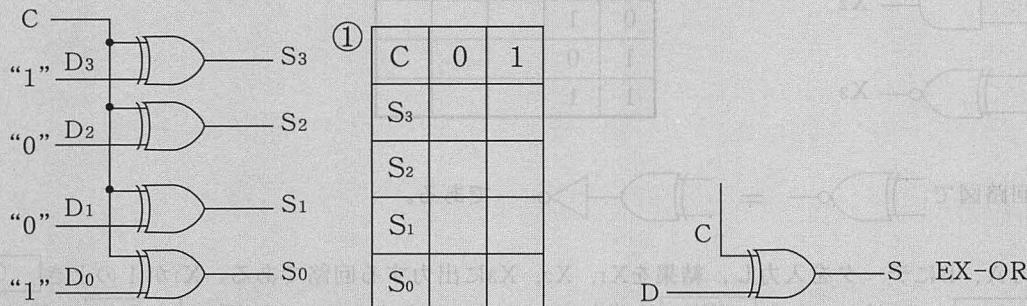
- ①ウ ②ア ③イ

例題2.6 図の論理回路において、入力D(1001)のとき、(第30回)

(1) コントロール信号Cによって出力S₀～S₃はどうなるか、真理値表①を完成させなさい。

(2) この論理回路の説明について文中の空欄に入る最も適切な語を解答群から選びなさい。

この回路は、右の図のようなEX-OR回路を組み合わせて構成されている。2入力のEX-OR回路の真理値表は、②であるので、入力Cが0のとき、③が出力され、入力Cが1のとき、④が出力される。したがって、入力Cによって、③か④か選択できる4ビットの⑤回路であるといえる。



解答群

- | | | | |
|---------------------------|-------------------------|---------------------------------|----------------------------------|
| ア. 入力Cの値
カ. 2の補数
コ. | イ. 入力Dの値
キ. 比較
サ. | ウ. 入力Cの値の反転
ク. マルチプレクサ
シ. | エ. 入力Dの値の反転
ケ. デマルチプレクサ
ス. |
|---------------------------|-------------------------|---------------------------------|----------------------------------|

入力		出力
D	C	S
0	0	1
1	0	0
0	1	0
1	1	1

入力		出力
D	C	S
0	0	0
1	0	1
0	1	1
1	1	0

入力		出力
D	C	S
0	0	0
1	0	1
0	1	0
1	1	1

入力		出力
D	C	S
0	0	1
1	0	0
0	1	1
1	1	1

解説

各ビットのEX-OR回路は、Cの入力が1のとき、1の補数を出力できる。また、Cの入力が0のときには入力されたデータがそのまま出力される。4ビット分のCが接続されているので、4ビットの1の補数を出力する。

解答

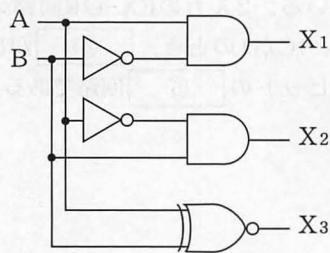
(1) ①

C	0	1
S ₃	1	0
S ₂	0	1
S ₁	0	1
S ₀	1	0

(2) ②サ ③イ ④エ ⑤オ

練習問題

7 次の論理回路の真理値表を完成して、文中の [] に入るべき語を解答群から選びなさい。(第28回)



入力		出力		
A	B	X ₁	X ₂	X ₃
0	0			
0	1			
1	0			
1	1			

ただし、論理回路図で = である。

この回路は入力A, Bにデータを入力し、結果をX₁, X₂, X₃に出力する回路である。X₁が1のとき [②]、X₂が1のとき [③]、X₃が1のとき [④] を表している。これらより、この回路は [⑤] 回路であるといえる。

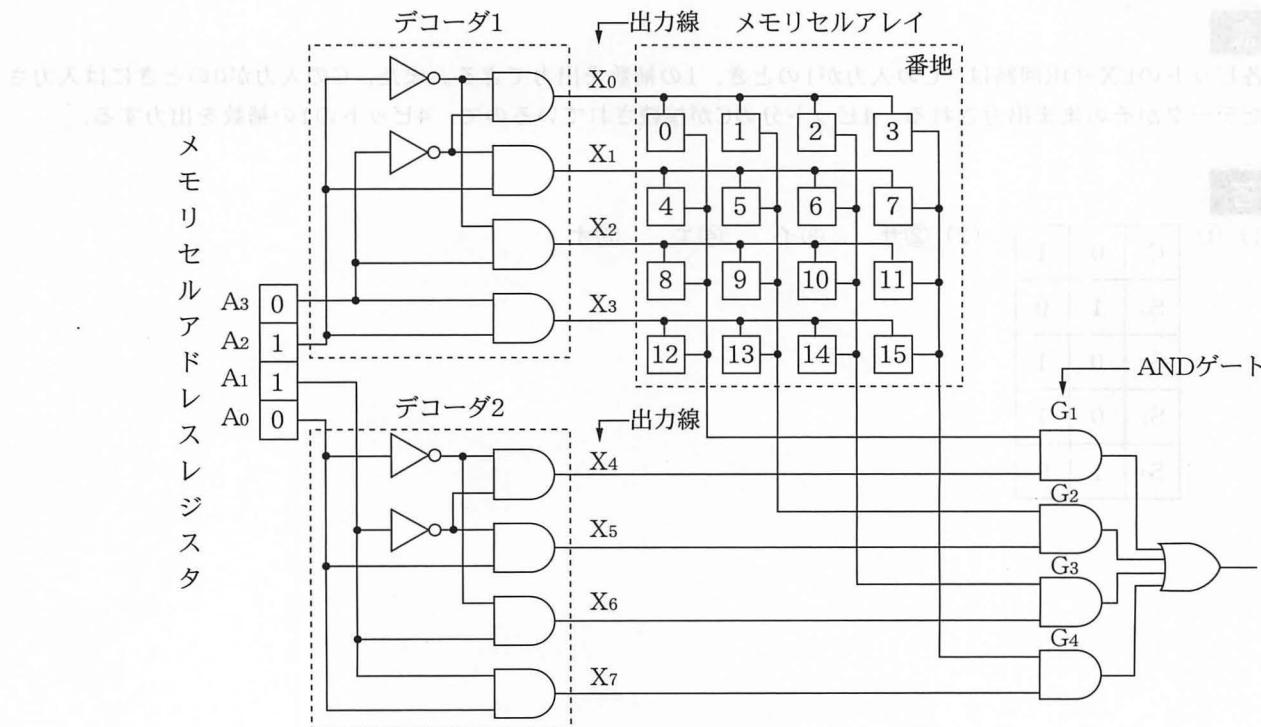
解答群

- ア. A=B イ. A<B ウ. A>B エ. A≤B オ. A≥B ハ. 加算回路 キ. 比較回路
- ク. 一致回路

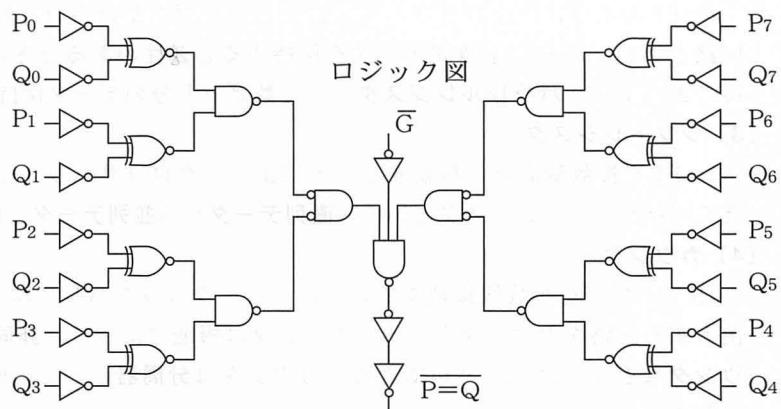
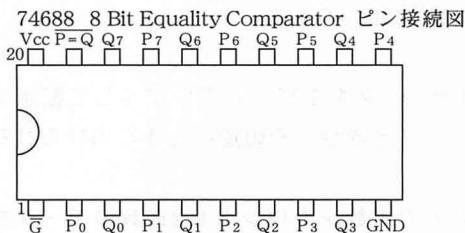
8 図は16ビットメモリセルの読み出し回路である。次の文章中の [] に当てはまる図中の記号や数値を入れなさい。(第34回)

図のメモリセルアドレスレジスタに0110が入力されたとき、上位2ビットによってデコーダ1の出力線は、[①] が1になり選択される。

また、下位2ビットによって、デコーダ2の出力線は [②] が1になり選択されるので、ANDゲート [③] を通じてメモリセルアレイの [④] 番地の記憶内容が読み出される。



- 9 下図のTTL ICにおいて、 $P_0 \sim P_7$, \bar{G} , $\bar{P} = \bar{Q}$ がそれぞれ下表のとき、 $Q_0 \sim Q_7$ はどうなるか。0または1を埋めて表を完成しなさい。(第33回)



ただし、EX-NORはEX-ORとNOTの組み合わせである。

また、次のような関係がある。

$$\text{AND-Not} = \text{NOT-AND} = \text{Inverter} = \text{NOT-Inverter} = \text{NAND-Not}$$

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	\bar{G}	$\bar{P} = \bar{Q}$
1	0	1	0	1	1	0	0	0	0

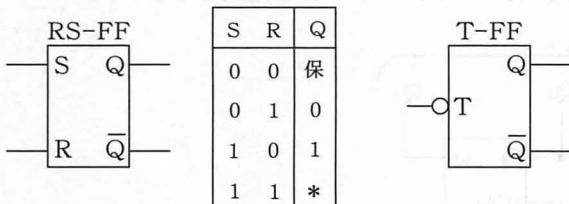
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0

2. 3 フリップフロップ(FF) を応用した回路

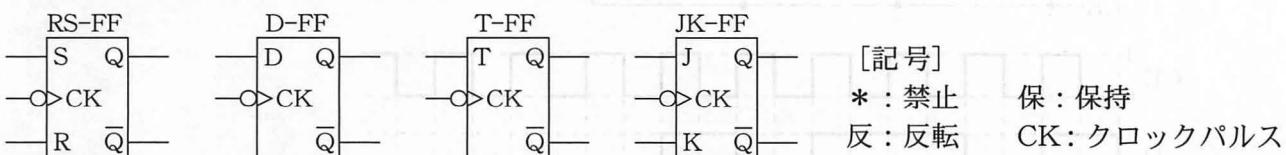
要 点

[1] フリップフロップの基本動作

- ・非同期：入力によって即座に出力が変化する。



- ・同期：入力の状態によりクロック信号で出力が変化する。



CK	S	R	Q
↓	0	0	保
↓	0	1	0
↓	1	0	1
↓	1	1	*

CK	D	Q
↓	0	0
↓	1	1

CK	T	Q
↓	0	保
↓	1	反

CK	J	K	Q
↓	0	0	保
↓	0	1	0
↓	1	0	1
↓	1	1	反

→ ローアクティブ
(0のとき信号が有効)

→ ネガティブエッジトリガ
(立下りのとき動作)

(2) ラッチ

D-FFのように、クロックパルスによりトリガがかけられた瞬間のデータを保持(記憶)する回路をラッチ回路といい、ある一定時間データを保持する必要性のあるときに用いられる。D-FFを複数個並列に接続することにより、パラレルレジスタとして数ビット分のデータ保持(記憶)に用いられる。

[3] シフトレジスタ

D-FFを複数個直列に接続することにより、クロックパルスで直列データを1ビットずつずらして転送できる回路をシフトレジスタといい、直列データから並列データを取り出すときや、その逆のときに用いられる。

[4] カウンタ

JK-FFなどを複数個接続することにより、クロックパルスなどの入力の数をカウントし2進数のデータを出力する回路をカウンタといい。カウンタは機能によって、**非同期式**、**同期式**、**アップカウンタ**、**ダウンカウンタ**などに分けることができる。カウンタは分周器としての働きもあり、1段通過するごとに周波数は $\frac{1}{2}$ になる。

□例題2.7□ 図は4ビットパラレルレジスタである。

下の機能に適する端子記号を選びなさい。

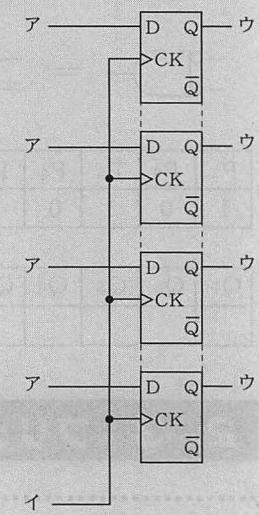
- ①出力端子
- ②データ入力
- ③入力同期信号

解説

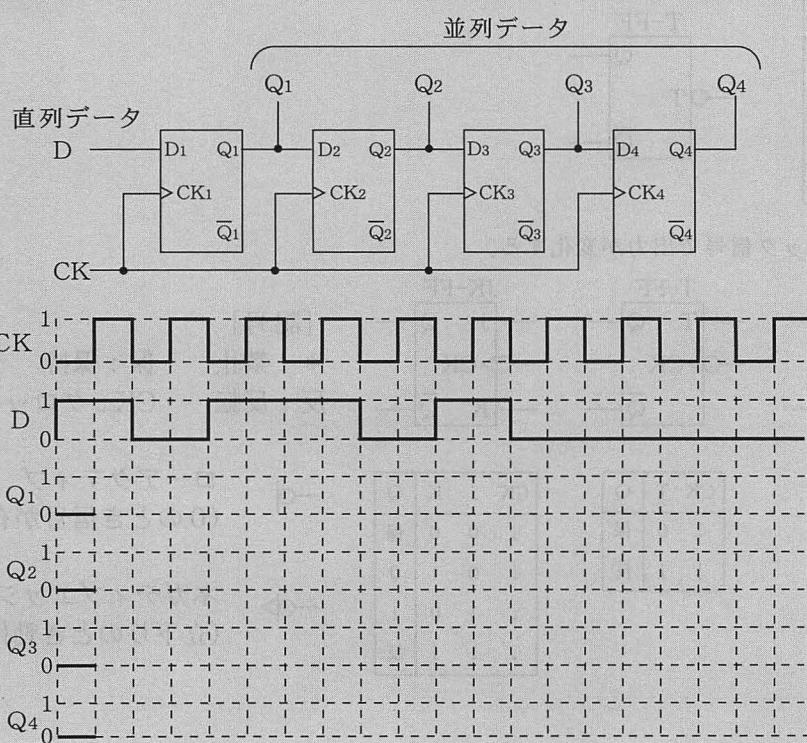
パラレルレジスタはD-FF回路を図のように並列に接続し、データの保持を行う回路である。図のD-FFのCK入力は図よりポジティブエッジ(立ち上がり)で動作することがわかる。(ア)のそれぞれのビットに入力されるデータを、(イ)から入力されるクロックパルスのポジティブエッジ(立ち上がり)により保持(記憶)し、その保持(記憶)したデータを(ウ)から出力する。データラッチとも呼ばれる。

解答

- ①ウ
- ②ア
- ③イ



□例題2.8□ 図のような4ビットのシフトレジスタがある。次のタイムチャートを完成しなさい。



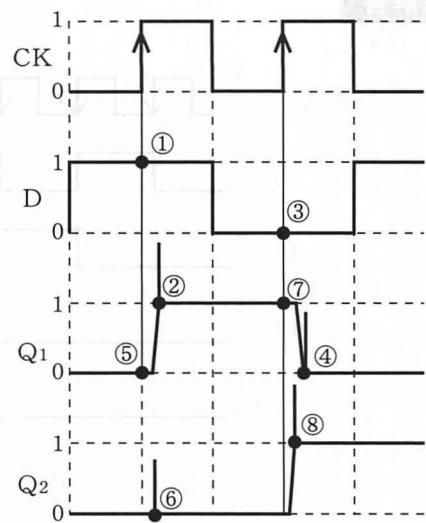
解説

シフトレジスタはD-FF回路を図のように直列に接続し、データの直並列変換をおこなう回路である。図のD-FFのCK入力は図よりポジティブエッジ(立ち上がり)で動作することがわかる。

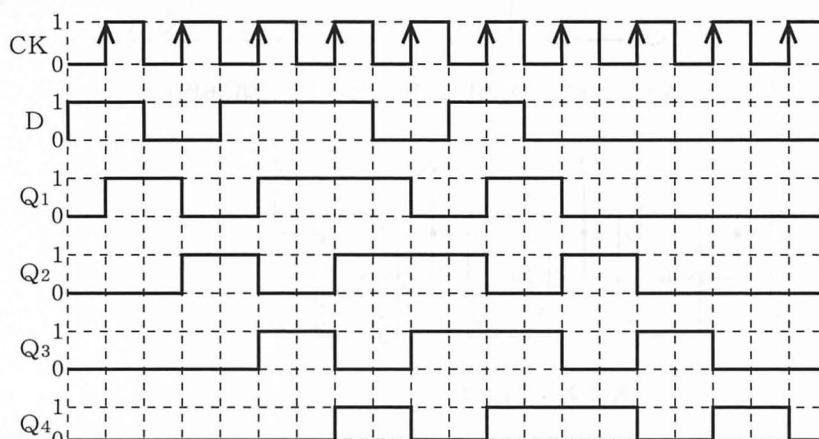
- ①CKの立ち上がりでDのデータ1を得る。
- ②データ1をQ1に出力するが、実際には微少な時間遅れを生じる。
- ③CKの立ち上がりでDのデータ0を得る。
- ④データ0をQ1に出力するが、実際には微少な時間遅れが生じる。
- ⑤CKの立ち上がりでQ1のデータを調べると、②の時間遅れのため、データは0となる。
- ⑥データ0を出力する。
- ⑦CKの立ち上がりでQ1のデータを調べると、④の時間遅れのため、データは1となる。
- ⑧データ1を出力する。

以下同様に、CKの立ち上がりでQ1～Q4はそれぞれ1クロックずつ遅れて出力される。すなわち、1クロックごとにデータが右へシフトしてゆく。

ただし、解答のように、時間遅れについては簡易なタイムチャートには表現されない場合が多いので注意が必要である。



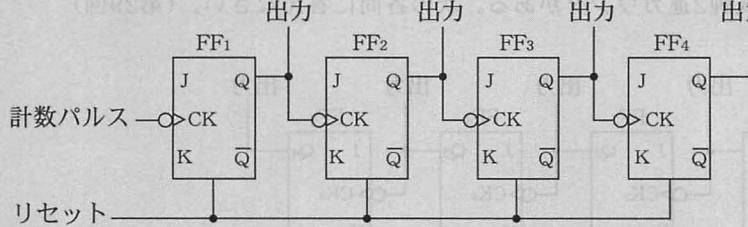
解答



□例題2.9□

図のような4ビットの非同期式純2進カウンタがある。次の間に答えなさい。

(ただし、入力J, Kは両方とも1である)



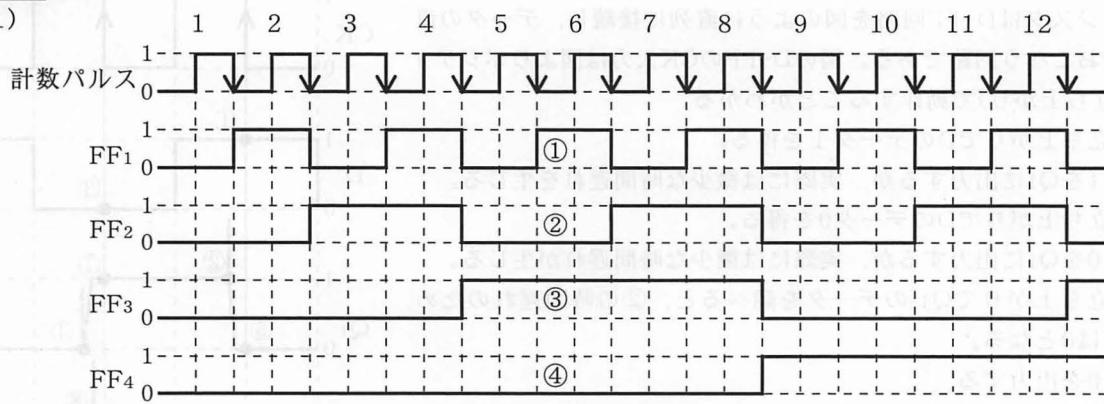
- (1) 各FFのタイムチャートを示しなさい(カウンタは最初リセット状態にあるものとする)。
- (2) 5個目の計数パルスで作動したとき、各FFはどのような状態となるか。“0”，“1”で示しなさい。

解説

- (1) 入力パルスの立下りで動作する。
- (2) タイムチャートの①～④の状態を読みばよい。

解答

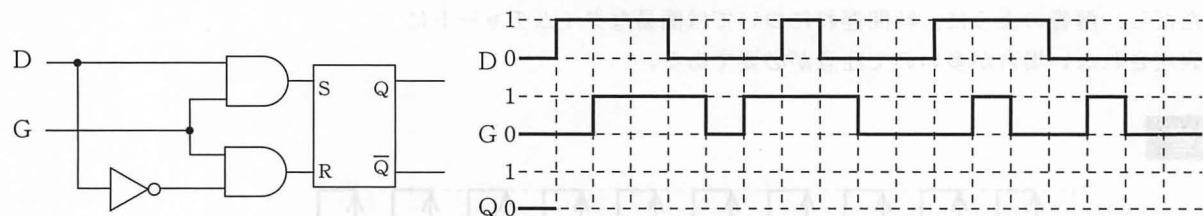
(1)



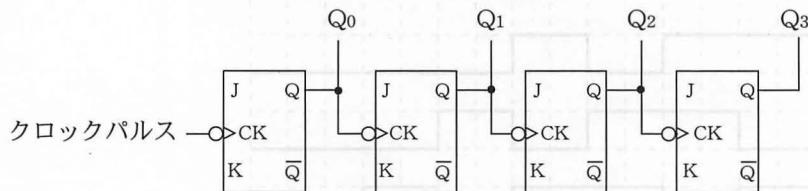
(2) $FF_1 : 1 \quad FF_2 : 0 \quad FF_3 : 1 \quad FF_4 : 0$

練習問題

10 図のRS-FFの回路において。タイムチャートを完成しなさい。(第36回)



11 図に示す4ビット非同期式の2進カウンタについて、次の間に答えなさい。(第36回)

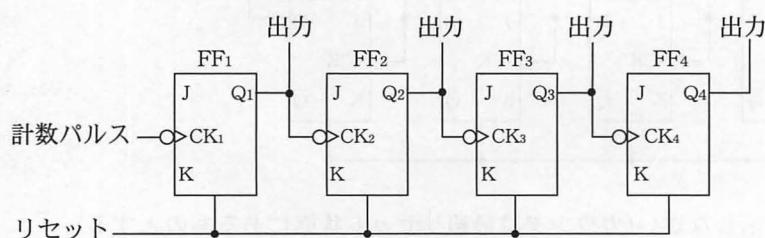


J, K はすべて1とする

①1段目に繰返し周波数80MHzのクロックパルスが入ったとき出力 Q_3 は何MHzになるか。

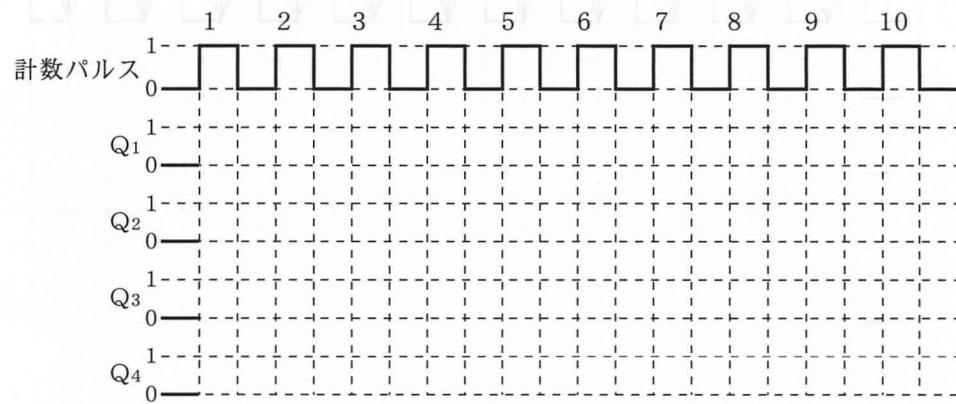
②初め、 $Q_0=0, Q_1=0, Q_2=0, Q_3=0$ であるとすると、クロックパルスが5個入ったとき、 $Q_0 \sim Q_3$ はどうなるか。

12 図のような4ビットの非同期式純2進カウンタがある。次の各間に答えなさい。(第29回)



J, K はすべて1とする

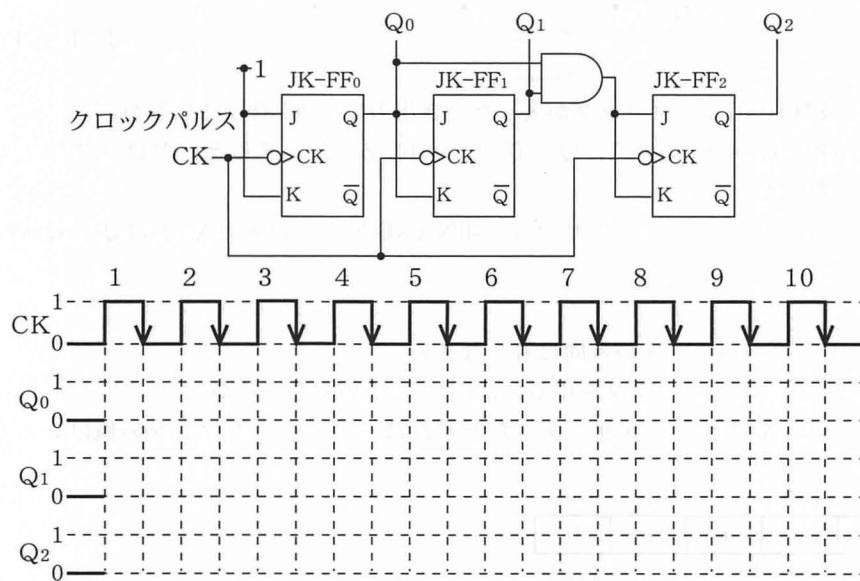
(1) 各FFのタイムチャートを示しなさい。(FFは入力パルスの立下りで動作し, カウンタは, 最初リセット状態にあるものとする)。



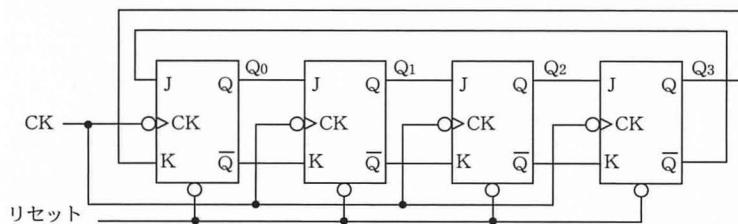
(2) 5個目の計数パルスで作動した直後, 各FFの出力Q₁, Q₂, Q₃, Q₄はどのような状態となるか。それについて, 0または1で答えなさい。

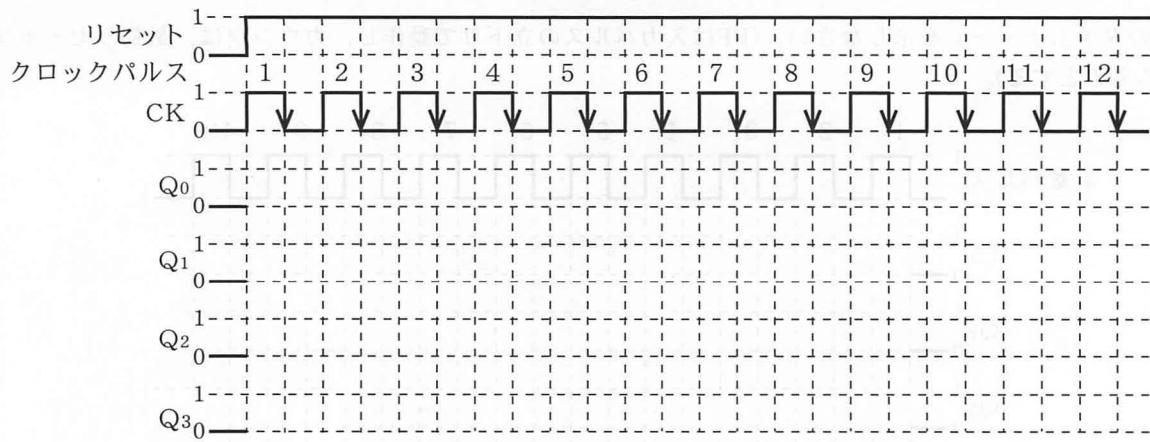
13 図の回路は同期式8進カウンタである。各FFのタイムチャートを示しなさい。

ただし, JK-FF0の入力J, Kは両方とも1である。(第30回, 第34回)

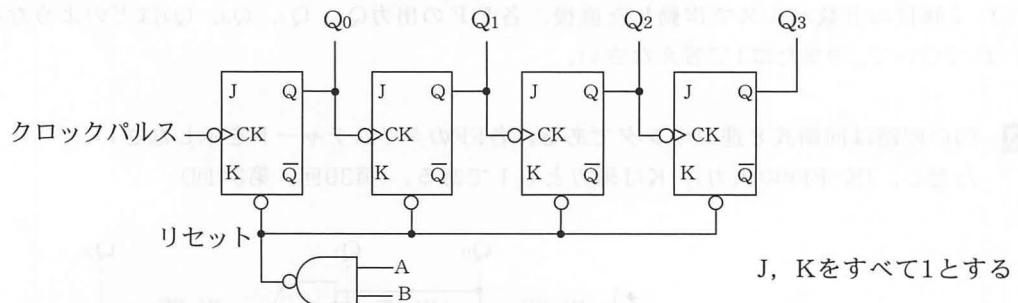


14 下図のJK-FFを4個接続したジョンソンカウンタと呼ばれる回路において, リセット信号で図のようにQ₀～Q₃が0になった後のタイムチャートを描きなさい。(第32回)





15 図で示す4ビット非同期式の2進カウンタについて、次の各間に答えなさい。



- ①1段目に繰返し周波数10MHzのクロックが入ったとき出力Q₃は何MHzになるか。
- ②カウンタの内容が初めQ₀=0, Q₁=1, Q₂=0, Q₃=0であったとすると、クロックパルスが4個入ったときその内容はどうなるか。
- ③これを10進カウンタとするためには、リセット用NANDゲートの入力A, BはQ₀～Q₃のどこに接続すればよいか。

16 純2進カウンタ(T-FF)について、次の各間に答えなさい。

- ①500までの数をカウントするカウンタを作るにはFFを何個必要とするか。
- ②5ビットのカウンタの各FFが図の状態にあったとすればこのときのカウンタの値はいくらか、10進数で答えなさい。

FF ₄	FF ₃	FF ₂	FF ₁	FF ₀
1	0	1	1	0

MSB LSB

- ③水晶発振器で1.048576MHzの信号を発生させて18段のT-FFを通過後の周波数は何Hzか。

2.4 エンコーダ・デコーダとマルチプレクサ・デマルチプレクサ

要 点

デジタル回路で使われる数はすべて0と1の組合せの2進数である。しかし、我々は日常10進数を扱っている。10進数データなどをコンピュータの信号である2進数に変換する回路などをエンコーダ(コード化する), 反対をデコーダという。

①エンコーダ

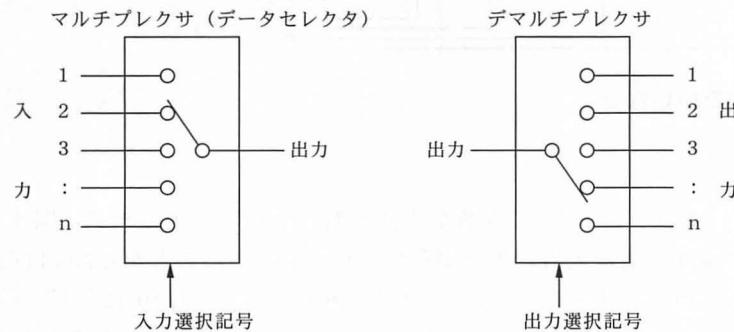
- ・10進数→BCDコード
- ・16進数→4ビット2進数
- ・キー入力変換

②デコーダ

- ・BCDコード→10進数
- ・4ビット2進数→16進数
- ・BCDコード→7セグメントLED信号

③マルチプレクサ, デマルチプレクサ

信号を選択処理する回路に図に示すようなマルチプレクサ, デマルチプレクサがある。

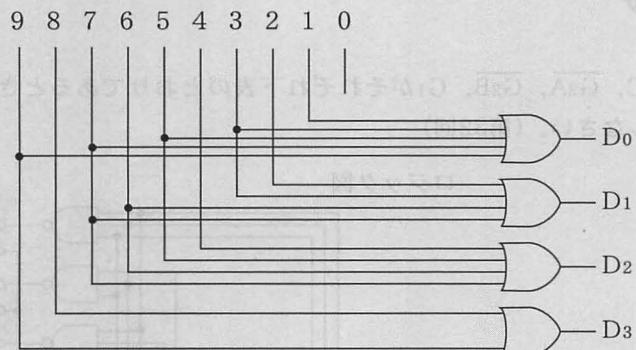


7セグメントLED

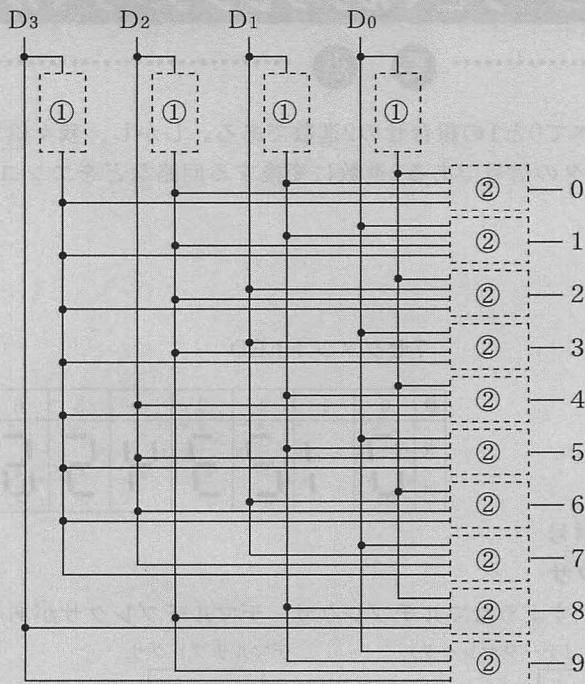
数	0	1	2	3	4	5	6	7	8	9
表 示	a f e d b	b c e d g	a b c d g	a b c d g	f b c d g	f b c d g	f b c d g	f b c d g	f b c d g	f b c d g

□例題2.10□ 10進数と2進数の変換を行う回路について次の各間に答えなさい。

(1) 次の回路のはたらきをいいなさい。



(2) 次の回路は(1)の反対の処理を行う回路である。①, ②にあてはまる論理回路は何か。



(3) (2)の回路のはたらきをいいなさい。

解説

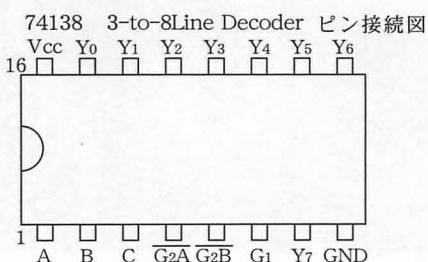
10進数から2進数に変換するエンコーダ、2進数から10進数に変換するデコーダに関する問題である。デコーダの回路の①にあてはまる論理回路は入力と出力がそれぞれ一つであることからNOT回路であることが簡単に予想できる。②については、例えばD₀～D₃のそれぞれが0000のとき、出力0に”1”を出力させるためには、AND回路を用いるとよい。以下同様にD₀～D₃のそれぞれが0001, 0010, …を確認すればよい。

解答

- (1) 10進数を2進数に変換するエンコーダ
- (2) ①NOT回路 ②AND回路
- (3) 2進数を10進数に変換するデコーダ

練習問題

17 下図のTTL ICにおいて、A, B, C, $\overline{G_2A}$, $\overline{G_2B}$, G₁がそれぞれ下表のとおりであるとき、Y₀～Y₇はどうなるか。0または1を埋めて表を完成しなさい。(第32回)

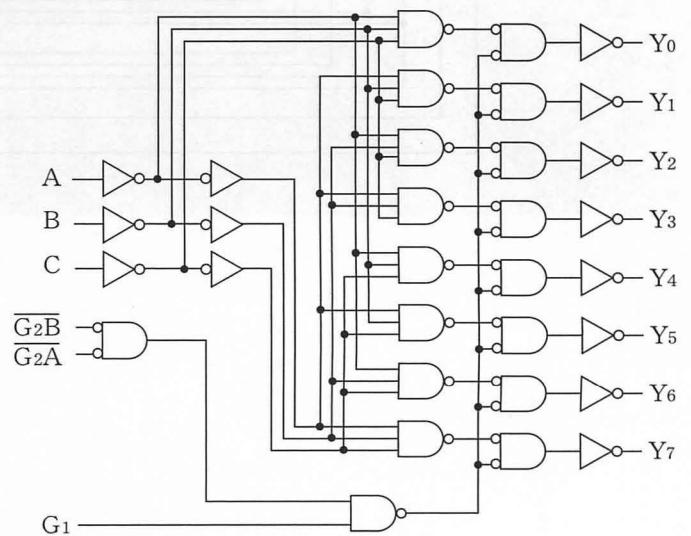


ただし、

$$\begin{array}{c} \text{---} \\ \text{O} \end{array} = \begin{array}{c} \text{---} \\ \text{O} \end{array}$$

$$\begin{array}{c} \text{---} \\ \text{O} \end{array} = \begin{array}{c} \text{---} \\ \text{O} \end{array}$$

ロジック図



A	B	C	$\overline{G_2A}$	$\overline{G_2B}$	G1
0	1	1	0	0	1

Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

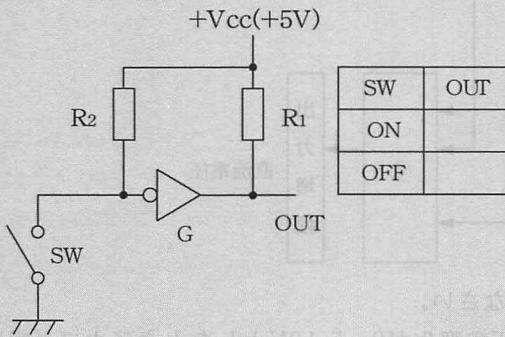
2.5 インタフェース回路

要 点

コンピュータがデータや信号をやりとりするときには、必ず信号形態を合わせる必要がある、その両者を合わせる部分をインターフェースといい、一般には電気的なことを指す。インターフェース回路を知るためにには、ICの規格、各種素子、変換回路、通信規格などいろいろな知識を理解することが大切である。

□例題2.11□

図でSWを開じたとき、R₂に流れる電流の向きを矢印で示しなさい。また、OUT端子の電圧とSWのON、OFFとどのような関係にあるか、1(+5V)、0(0V)で答えなさい。



解説

電流は電位の高い方から低い方に流れる。この場合は上から下。出力電圧はSWがONのときはIC入力は0であるから出力は1、OFFのときは反対である。

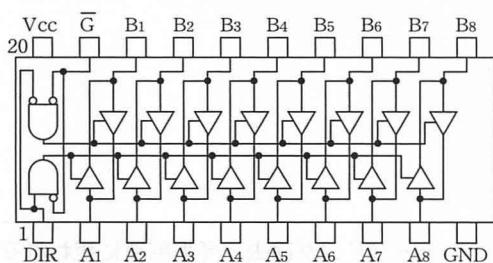
解答

↓（下向き）、ONのとき1、OFFのとき0

練習問題

- 18 図のTTL ICの真理値表を完成しなさい。ただし、①～⑤には、0または1のいずれかの論理が入るものとする。ただし、スリーステートバッファ回路は、出力が、ゲート入力により三つの状態をとる。ゲートが1のとき入力(1または0)がそのまま出力され、ゲートが0のとき入力に関係なく出力はハイインピーダンス(出力線が接続されていないのと同じ)になる。(第31回)

74245双方向スリーステートバッファ

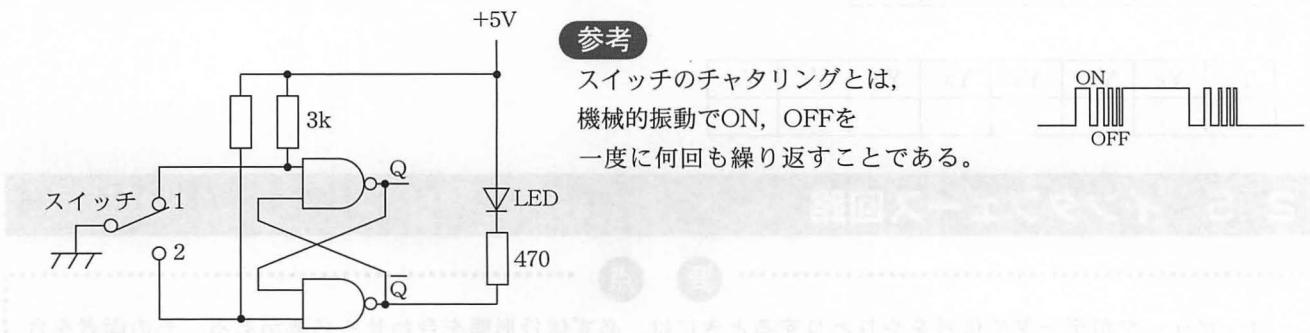


制御信号		信号の流れ
\overline{G}	DIR	
①	②	B → A
③	④	A → B
⑤	不問	絶縁

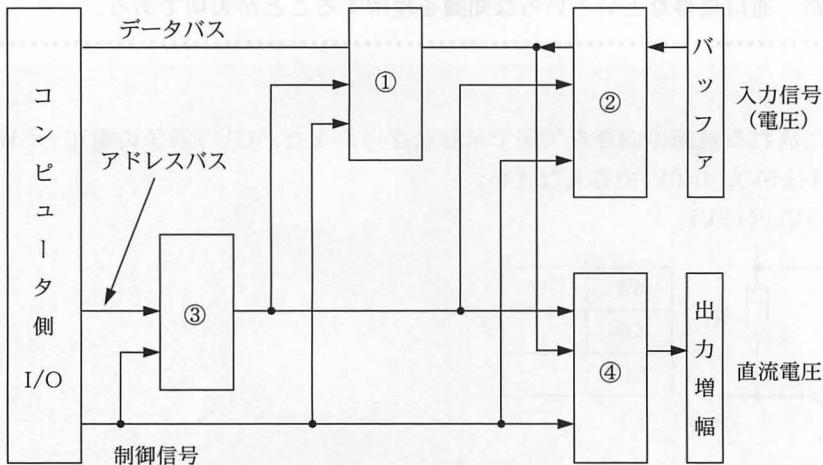
スリーステートバッファ回路



- 19 NANDのICを2個使用して、スイッチ回路を作った。この回路の役割は何か。スイッチが1側のとき、Q, \bar{Q} の出力に電圧1(+5V), 0(0V)を記入しなさい。また、このときLEDは点灯するか。



- 20 下図は、コンピュータによるアナログ電圧の入出力を示すブロック図である。下記の間に答えなさい。



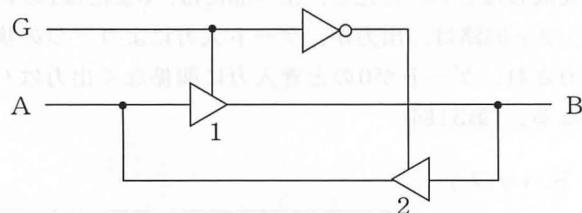
- (1) 図中の①～④までの機能は何か、解答群から選びなさい。
- (2) コンピュータのデータが8ビットで、アナログ電圧の変化が0～5.12Vとしたときアナログ電圧の最小分解能はいくらか。

解答群

- ア. A-D変換 イ. D-A変換 ウ. バスバッファ エ. アドレスデコーダ

- 21 図について間に答えなさい。

- ①図で、端子Gを論理レベル“0”にすると、回路1, 2のどちらが信号を通過できる状態になるか。
- ②前問に続いて端子Gを論理レベル“1”としたとき、端子A, Bのどちらから入れた信号が、この回路を通過できるか。



- 22 次のICの特長を説明する文を下から選びなさい。

- ①オープンコレクタIC
②スリーステート形IC
③シュミットトリガIC

解答群

- ア. 入力感度のヒステリシスがありノイズ対策に利用したりスイッチのチャタリング除去、発振回路に使われる。
イ. 出力電圧電流の増加、バスへの接続にワイヤードオアができる。
ウ. 出力のハイインピーダンス状態がありバスとの接続に使う。

3. コンピュータの基本構成と各部の働き

3. 1 基本構成

要 点

[1] コンピュータの基本構成

現在のほとんどのコンピュータが、フォン・ノイマンが提唱した蓄積プログラム制御方式を採用しており、それらはノイマン型コンピュータと呼ばれている。

①プログラム内蔵方式(stored program)

プログラムを主記憶に格納しておき処理を実行する方式

②逐次制御方式(sequential control)

主記憶装置の命令を制御装置に取り出し、解読し、順次処理を行う方式

③五大機能

入力、出力、記憶、演算、制御

これらの機能を実現するための装置が、入力装置、出力装置、記憶装置、演算装置、制御装置の五つで、五大装置と呼ばれる。

[2] 制御装置

制御装置は、主記憶装置内のメモリに格納されているプログラムの命令語を順次読み出して命令解読器(命令デコーダ)で解読し、算術論理演算装置や入出力装置などに順序正しく制御指令を与える。制御装置には次のようなレジスタやデコーダが用意されている。

①命令レジスタ(instruction register)

主記憶装置から読み出した命令語を一時的に記憶するレジスタ。

②命令デコーダ(instruction decoder)

命令レジスタの命令語から命令コード部を取り出して解読する回路である。

③指標レジスタ(index register)

アドレスを修飾する場合に、数値を記憶するレジスタでインデックスレジスタともいう。

④基底アドレスレジスタ(base address register)

プログラムやデータの基準アドレスを記憶するレジスタで、命令部のアドレスを基準アドレスからの変位として加算して実行アドレスを算出する。ベースアドレスレジスタともいう。

⑤命令アドレスレジスタ(instruction address register)

次に実行する命令の主記憶装置のアドレス(ロケーション)を示すレジスタで、プログラムカウンタなどとも呼ばれる。

[3] 演算装置

演算装置は制御装置からの指示により、算術論理演算装置を用いて算術演算や論理演算などを行う。演算結果を一時的に保持しておくアキュムレータ(accumulator, 累算器)や、演算の結果オーバーフローや正・負・ゼロになったかなどを記憶しておくフラグレジスタなどのレジスタから構成される。

[4] 命令の実行

入力装置や補助記憶装置などから主記憶装置に格納されたプログラムは、二つの動作、すなわち、命令の読み取り段階と命令の実行段階が繰り返されて処理される。

①命令取り出し段階(フェッチサイクル)

1.命令アドレスレジスタの内容が、メモリアドレスレジスタに転送される。

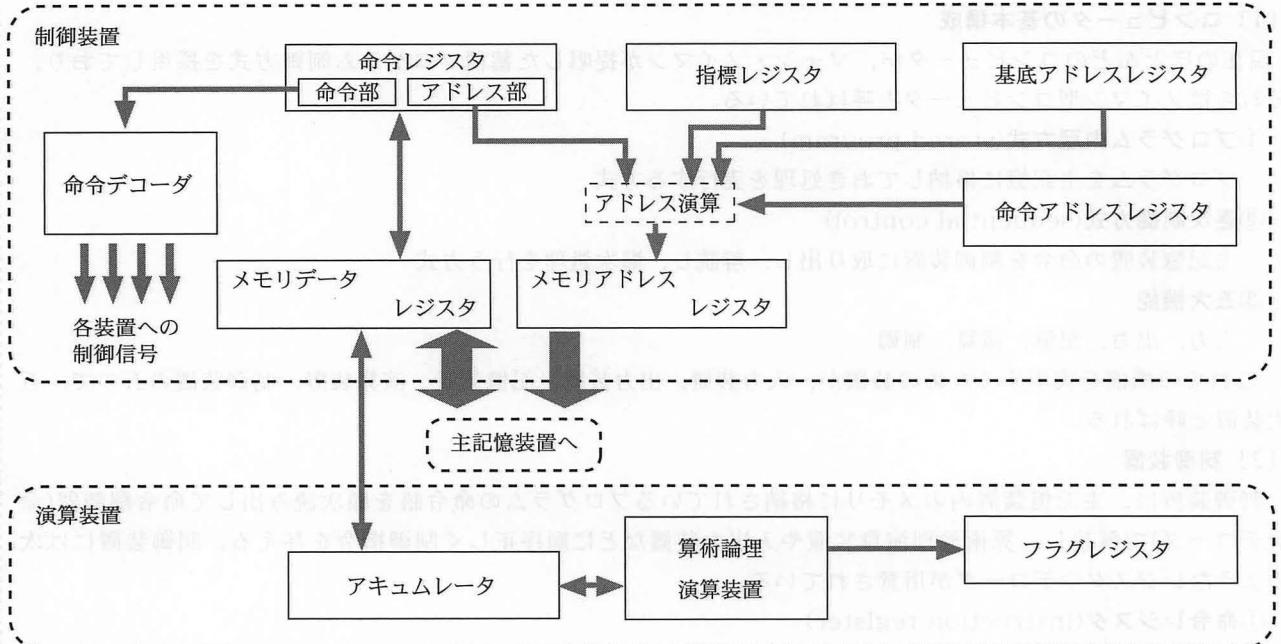
2.命令アドレスレジスタの内容に、実行した命令の長さが加えられる。

3.メモリアドレスレジスタで選択されたアドレスの内容が、メモリデータレジスタに読み取られる。

4.メモリデータレジスタの内容が解読のため命令レジスタに移される。

②命令実行段階(エグゼキューションサイクル)

- 命令レジスタの命令部の内容が、命令デコーダに移されて解読される。
- 命令デコーダで解読された結果から、必要な回路や装置に指令が出される。
- 命令レジスタのアドレス部、指標レジスタ、基底アドレスレジスタの値から操作対象となるアドレスが計算され、メモリアドレスレジスタに転送される。
- メモリアドレスレジスタで指定されたアドレスの内容がメモリデータレジスタに読み出される。
- メモリデータレジスタのデータとアキュムレータのデータとが演算され、結果がアキュムレータに入る。



[5] アドレスの指定方式

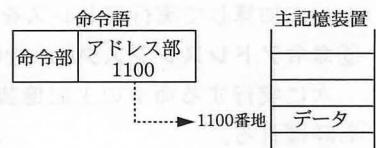
① アドレスの表し方

主記憶装置の各記憶場所に割り当てられた固有のアドレスを絶対アドレスという。また、基準のアドレスからどれだけ離れた場所かを示す方法を相対アドレスという。命令実行のとき、実際に読み取り・書き込みをする主記憶装置のアドレスを有効アドレスという。

② アドレスの指定

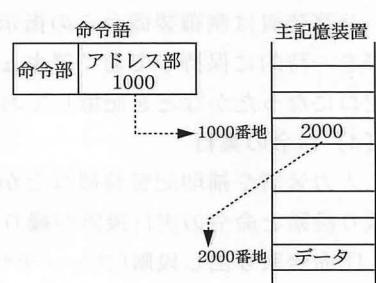
1. 直接アドレス指定

命令語のアドレス部の値を有効アドレスとして主記憶装置のアドレスを指定する。
例えば、アドレス部が1100であれば有効アドレスは1100番地となる。



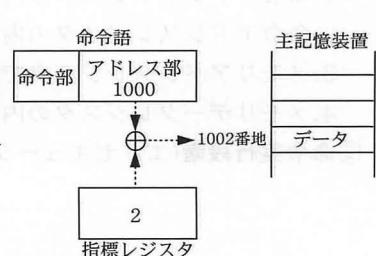
2. 間接アドレス指定

命令語のアドレス部で指定された主記憶装置の内容を有効アドレスとして指定する。
例えば、アドレス部を1000とし、指定された1000番地の内容を2000とすれば、有効アドレスは1000番地の内容すなわち2000番地となる。



3. 指標アドレス指定

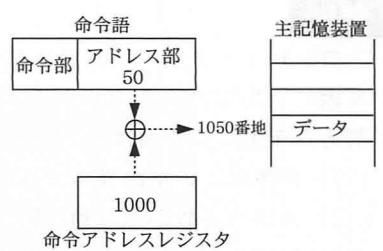
命令語のアドレス部と指標レジスタの内容を加えた値を有効アドレスとする。
例えば、アドレス部が1000、指標レジスタの内容が2であるとき、有効アドレスは1002番地となる。



4. 相対アドレス指定

命令語のアドレス部で何番地前後するのかを相対的に指定した値を有効アドレスとする。

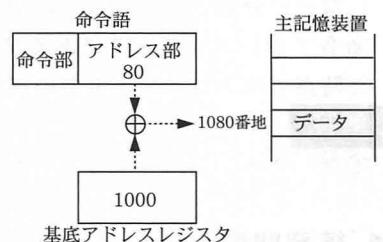
例えば、命令語のアドレス部が50、命令アドレスレジスタが1000のとき、1000番地より50離れた1050番地が有効アドレスになる。分岐命令などに用いられる。



5. 基底アドレス指定

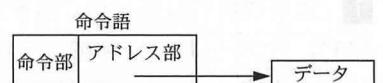
命令語のアドレス部と基底アドレスレジスタの内容を加えた値を有効アドレスとする。

例えば、命令語のアドレス部が80、基底アドレスレジスタが1000のとき、1000番地より80離れた1080番地が有効アドレスになる。再配置可能なプログラムなどに用いられる。



6. 即値アドレス指定

命令語のアドレス部の値を演算データとして利用する方式で、主記憶装置へのアクセスとは直接的には関係ない。



□例題3.1□ 次の説明に最適な語句を解答群から選びなさい。

- ①コンピュータの制御装置の一部であって、記憶装置から読み取られた命令を受け取り、それを実行するために一時記憶しておくレジスタ。
- ②演算のための主要なレジスタであって、四則演算、論理演算などの結果を一時記憶しているもの。
- ③命令実行の際、命令語のアドレス部に、このレジスタの値を用いてアドレス修飾を行うことがある。このアドレス修飾に用いるレジスタ。
- ④主記憶装置と制御装置の間にあり、命令語やデータを読み書きする際には必ず経由するレジスタ。

解答群

- | | | | |
|---------------|------------|---------------|-----------|
| ア. エンコーダ | イ. 命令レジスタ | ウ. メモリデータレジスタ | エ. レジスタ |
| オ. アキュムレータ | カ. 入出力チャネル | キ. スタック | ク. 指標レジスタ |
| ケ. 命令アドレスレジスタ | | | |

解説

レジスタには、CPU内部で演算などの途中の処理などに必要な小さなサイズのデータを一時的に記憶するはたらきがある。

解答

- ①イ ②オ ③ク ④ウ

□例題3.2□ 次に示す文の空白に入るべき用語を解答群から選び出し、その記号を [] に記入しなさい。

割込み処理の命令制御動作は、次のように行われる。まず、①は、各命令実行段階の終了ごとに割込み要求があるかどうかを調べて、要求があれば次の②を行うことを中断し、その割込み処理に対応する処理ルーチンを実行する。割込みによって中断されたプログラムは、③終了後継続して実行されなければならないから、割込みが生じたときの④の内容は、所定の番地やスタックなどに待避させておくようになっている。割込み受け後、その処理プログラム実行中に他の⑤が発生する場合がある。このとき現在実行中の処理を、より緊急度の低い割込みによって中断されたくない場合、いくつかある方法の一つとして、割込み原因の一部または全部を隠ぺいする方法がある、これを⑥という。

解答群

- | | | | | |
|---------|-------------|---------|-----------|---------------|
| ア. リセット | イ. 割込み禁止マスク | ウ. 割出し | エ. 割込み処理 | オ. 命令取出し |
| カ. 命令実行 | キ. 割込み要求 | ク. 制御装置 | ケ. 命令レジスタ | コ. 命令アドレスレジスタ |

解説

コンピュータが一つのプログラムを実行中、内的あるいは外的要因によって実行中のプログラムを一時中断し、要因に対応した別のプログラムの実行に移ることを割込みといい、ハードウェア上の重要な機能の一つである。割込みを生じる要因は、

- (1) 停電などの電源異常や装置の一部に異常が生じる等のハードウェア割込み
- (2) 入出力装置からの動作完了時やタイマからの信号等外部からの割込み信号による外部割込み
- (3) 制御装置内部で発生する内部割込み

などに分類され、これらの割込み制御は、制御装置が割込み処理プログラムによって行う。割込みが生じると、命令アドレスレジスタにあるアドレス値は、一時的にスタックと呼ばれるメモリに保存され、割込みが終了した時点で戻され、割込み前のプログラムが続行できるようになっている。

解答

- ①ク ②オ ③エ ④コ ⑤キ ⑥イ

練習問題

1 次の説明に最も適切な語を解答群から選びなさい。(第32回)

- (1) ① は、一時的にデータを記憶しておくため主記憶装置上の場所で、最後に記憶したデータを最初に取り出す ② と呼ばれる方法で利用される。レジスタの値を一時的に退避させたり、サブルーチンへ分岐させる際の戻り番地を入れたりする場合に用いられる。
- (2) ③ アドレス方式はアドレスの指定を、その命令の所在場所から±nの形で指定する。この方式は、④ の中で、プログラムを自由に配置し得る ⑤ なプログラムとすることができます。また、命令語で指定した場所に、その命令の作用する ⑥ が入っている方式を ⑦ アドレス指定方式といい、他のプログラムと ⑧ 等を行う際に必要なアドレス指定方式である。
- (3) アセンブラー言語には、一つまたは複数個の機械語に変換される ⑨ 命令と、記憶場所や実行開始アドレスなどの指示をする ⑩ 命令がある。後者は対応する機械語がないから、機械語には変換されない。

解答群

- | | | | | | |
|----------|-------------------|----------|-----------|---------|----------|
| ア. 主記憶装置 | イ. 擬似 | ウ. 結合・編集 | エ. アドレスの値 | オ. 実行可能 | カ. 再配置可能 |
| キ. スタック | ク. ラストイン・ファーストアウト | ケ. 相対 | コ. 間接 | | |

2 文中の空欄に入る最も適切な語を解答群から選びなさい。(第29回)

命令を実行する際の制御装置の動作は次のようになる。まず命令カウンタに入っている数値をアドレスとする主記憶装置内のロケーションを探し、自動的にそのロケーションの内容が ① にロードされる。この段階までを命令取出し段階または ② という。次に、制御装置は命令ワード(命令語)の内容を ③ で解読し、それを元にした制御指令で ④ が開始される。なお、プログラムカウンタ(命令アドレスレジスタ)の内容に1が加えられ、 ④ の終了となる。このように、第一段階すなわち ② の後には、必ず次の段階すなわち ④ が続き、両者を合わせて一つの ⑤ を構成する。

解答群

- | | | | |
|---------------|-----------|------------------|---------------|
| ア. メモリデータレジスタ | イ. エンコーダ | ウ. フェッチサイクル | エ. インデックスレジスタ |
| オ. マシンサイクル | カ. 命令デコード | キ. エグゼキューションサイクル | ク. 命令レジスタ |

3 次の説明に最も関連しているものを解答群から選びなさい。

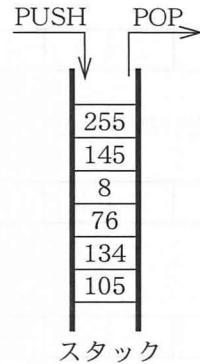
- ①CPUやメモリなどの装置が共有してアドレスやデータなどの信号を伝送する。
- ②CPUのかわりに周辺装置を制御する機能をもつ集積回路。
- ③CPU、メモリなどの間のデータ転送を制御する集積回路。
- ④コンピュータ本体にプリンタやディスプレイを接続するコネクタ。

解答群

- | | | | | |
|--------|----------|-----|--------|-------|
| チップセット | キャッシュメモリ | ポート | システムバス | 周辺LSI |
|--------|----------|-----|--------|-------|

- 4 次の順で10個の命令を実行したとき、スタックに格納されているデータは図のようになった。1番目のPUSH命令(①PUSH)でスタックに格納されたデータはどれか。

①PUSH→②PUSH→③PUSH→④POP→⑤PUSH→
⑥PUSH→⑦POP→⑧POP→⑨PUSH→⑩PUSH



- 5 次の各問の説明文を読み、最も適切と思われるものを解答群から選びなさい。

- ①同期式コンピュータにおいて、各部の動作を合わせるために用いられる周期的なパルス。
②入力信号を受け取って、内容が一つずつ増加または減少するように構成されたもの。
③演算装置の一部で、下のけたから上けた上げを行わない2進数1けたの加算回路。
④2進数から10進数の出力を得る回路。
⑤処理装置で実行するプログラムの介在なしに、メモリと入出力機器の間で直接にデータを転送するもの。
⑥演算のための主要なレジスタであって、四則演算、論理演算などの結果を一時記憶しているもの。
⑦レジスタの内容を保持するために用意されているメモリ領域であり、順次情報を記憶し、後から記憶した内容を先に取り出す仕組みになっているもの。
⑧現在実行しているプログラムを中断して別なプログラムを実行し、これが終了すると元のプログラムの実行を再開する処理のこと。
⑨計算した結果の正、負、オーバーフローなどといった情報を保持するレジスタ。
⑩処理装置内のあるレジスタから他のレジスタへ、または主記憶装置のロケーションから処理装置のレジスタへ1ワードの情報を転送すること。

解答群

- | | | | | |
|------------|----------|------------|------------|------------|
| ア. 割込み | イ. ロード | ウ. フラグレジスタ | エ. スタック | オ. アキュムレータ |
| カ. データカウンタ | キ. デコーダ | ク. ハーフアダー | ケ. クロックパルス | コ. ルーチン |
| サ. DMA | シ. エンコーダ | ス. フルアダー | セ. リセットパルス | |

- 6 次の文章に最も関連する語句を解答群から選びなさい。

- ①次の実行すべき命令語がどこにあるかを示す値が置かれるレジスタ。
②論理演算をした結果が保持されるレジスタ。
③命令解読器の直前にあり、メモリから取り出された命令語が置かれているレジスタ。
④次にアクセスするスタックの番地の値が置かれるレジスタ。
⑤計算した結果の正、負、オーバーフローなどといった情報を保持するレジスタ。

解答群

- | | | | |
|------------|---------------|---------------|----------------|
| ア. アキュムレータ | イ. スタックポインタ | ウ. 命令アドレスレジスタ | エ. 命令レジスタ |
| オ. フラグレジスタ | カ. コントロールレジスタ | キ. 指標レジスタ | ク. メモリバッファレジスタ |

3. 2 中央処理装置

要 点

[1] 高速化技術

逐次制御方式では、命令を読み取り解読して実行後、次の命令の読みとりを行う。これに対して、先行制御方式やパイプライン方式では、実行が完了する前に次の命令の読みとりを行う。先行制御方式では、先の命令の命令実行段階命令と次の命令の取り出し段階とを重ね合せて、並行的な処理を行う。パイプライン方式では、先の命令を取り出してすぐに、次の命令を取り出し次々と処理を行う。

・逐次制御方式

1	2	3	4	5	6
---	---	---	---	---	---

1	2	3	4	5	6
---	---	---	---	---	---

・先行制御方式

1	2	3	4	5	6
---	---	---	---	---	---

...

1	2	3	4	5	6
---	---	---	---	---	---

1	2	3	4	5	6
---	---	---	---	---	---

・パイプライン方式

1	2	3	4	5	6
---	---	---	---	---	---

1	2	3	4	5	6
---	---	---	---	---	---

...

1	2	3	4	5	6
---	---	---	---	---	---

...

[2] MIPSとFLOPS

MIPS(million instructions per second)は、CPUの性能を表す方法のひとつで1秒間に何百万回の命令を処理できるかを表している。FLOPS(floating point number operations per second)は、1秒間に何回の浮動小数点数演算(実数計算)ができるかを表している。

[3] CISCとRISC

従来のほとんどのコンピュータがCISC(Complex Instruction Set Computer, シスク, 複合命令セットコンピュータ)と呼ばれ、高水準言語の実行などを効率的にするため、複雑な命令セットを持ち、高度な処理の可能なものである。命令の基本動作(加算, シフトなど)に対応するマイクロ命令を用意し、マイクロ命令の組み合わせであるマイクロプログラムで命令を実現する方式で、命令コードによりROMに記憶されたマイクロプログラムを取り出して命令を実行する。これに対して、RISC(Reduced Instruction Set Computer, リスク, 縮小命令セットコンピュータ)はCISCと比較して命令の種類を少なくして高速化したものであり、ワークステーションと呼ばれるコンピュータでよく用いられている。一命令の長さを一定にして(固定長命令), すべての命令の実行に要する時間を同一にすることにより、命令の取り出し, 解読, 演算などを並列に処理するパイプライン制御が円滑に流れ全体の処理速度が高速になる。また、布線論理制御方式といい、処理の高速化のためにハードワイヤードロジック(命令の基本動作ごとに応じた論理回路)を持っている。

□例題3.3□

次のようなパイプライン処理で命令を実行するコンピュータで、10個の命令を実行する実行時間は何n秒か。ただし、各ステップの実行時間はすべて10n秒とし、パイプライン処理の実行を乱す要因はないものとする。

ステップ	処理内容
1	命令コード部取出し
2	命令解読
3	アドレス部取出し
4	実行番地計算
5	データ取出し
6	演算実行

最初の命令	1	2	3	4	5	6
2個目の命令	1	2	3	4	5	6
3個目の命令	1	2	3	4	5	6
	⋮					

解説

最初の命令は6ステップで、2個目の命令以降、各命令につき1ステップ分ずつ処理時間が延長される。また、各ステップの実行時間はすべて10n秒である。

$$10 \times 6 + 10 \times (10 - 1) = 150$$

解答

150n秒

□例題3.4□

50MIPSの性能のコンピュータで、9000万個の命令のプログラムを実行した場合の処理時間は何秒か求めなさい。ただし、プロセッサの使用率を60%として、他の要因で処理時間が遅くなることはないものとする。また、1 MIPS(million instruction per second)は、1秒間に100万個の命令を実行できる能力を意味している。(第34回)

解説

50MIPS(MIPSはMillion Instruction Per Second : 100万命令／秒)は1秒に 50×10^6 の命令を実行できることを意味している。プロセッサの使用率が60%であることより、実際の処理は、

$$50\text{MIPS} \times 60 \div 100 = 30\text{MIPS}$$

となる。したがって、1秒間に実行できる命令の数は 30×10^6 となり、9000万個の命令を実行する処理時間は、

$$9000 \times 10^4 \div (30 \times 10^6) = 3\text{秒}$$

となる。

解答 3秒

□例題3.5□

命令A, 命令B, 命令C, 命令Dを次の順に実行するプログラムがある。各命令の実効に必要なクロックサイクル数(CPI : Clock Per Instruction)は次の表の通りである。

CPUのクロックを2[GHz]とするとき、このプログラムのCPU実効時間[n秒]を求めなさい。(第31回)

命令	CPI
命令 A	5
命令 B	10
命令 C	8
命令 D	2

(プログラム)

A→A→B→B→A→D→B→C→A→B→A→A→B→C→D

解説

このCPUのクロックfcが2GHzであることより、1クロックサイクル時間Tccは、

$$Tcc = \frac{1}{fc}$$

と表せるので、

$$Tcc = \frac{1}{2 \times 10^9} = 0.5 \times 10^{-9} = 0.5\text{n秒}$$

で、0.5 n秒となる。

このプログラムで必要なクロック数は、命令Aを6回、命令Bを5回、命令Cを2回、命令Dを2回実効するので、CPIの表より、

$$5 \times 6 + 10 \times 5 + 8 \times 2 + 2 \times 2 = 100\text{クロック}$$

となり、先に求めた1クロックサイクル時間が0.5 n秒であることより、

$$0.5 \times 100 = 50\text{ n秒}$$

となる。

解答 50 n秒

練習問題

7 次の説明に最も適切な語を解答群から選びなさい。(第33回)

CISCは、高水準言語の実行などを効率的にするために、①な命令セットを持ち、高度な処理が可能である。命令の基本動作(加算、シフトなど)に対応するマイクロ命令を用意し、マイクロ命令の組み合わせである②で命令を実現する方式で、命令コードによりROMに記憶された③を取り出して命令を実行する。

これに対して、RISCは、CISCと比較して命令の種類を④して高速化したものである。一命令の長さを一定にして(固定長命令)、すべての命令の実行に要する時間を同一にすることにより、命令の取り出し、解読、演算などを並列に処理する⑤制御が円滑に流れ全体の処理速度が高速になる。また、布線論理制御方式といい、処理の高速化のために⑥(命令の基本動作ごとに對応した論理回路)を持っている。

解答群

- ア. 複雑 イ. パイプライン ウ. マイクロプログラム エ. ハードワイヤードロジック オ. 少なく

8 次の各間に答えなさい。答は、ア、イ、ウ、エから選びなさい。(第29回)

①命令セットを基本命令に限定して制御のための回路を簡単にして、更に固定長命令などでパイプライン制御が有効に働くようにして処理速度を高速化したアーキテクチャはどれか。

ア.CISC イ.RAID ウ.RISC エ.HDSL

②次の記述でRISCの説明として適切でないものはどれか。

ア.命令語長をできるだけ統一している。

イ.マイクロプログラム技術により高速化している。

ウ.パイプライン処理のしやすいアーキテクチャである。

エ.ハードワイヤードロジック制御で高速化している。

9 表に示す性能のコンピュータの処理性能は約何MIPSか。

命令種別	実行速度(μ秒)	出現頻度(%)
整数演算命令	1.0	60
移動命令	4.5	20
分岐命令	5.0	20

10 トランザクション1件について平均して60万命令の実行が必要であるとき、プロセッサの性能が15MIPSで使用率を80%とすると、このコンピュータのトランザクションの処理能力(件/秒)はいくらか。

11 すべての命令を5ステップで完了するパイプライン制御のコンピュータで、ある処理を行ったところ240n秒かかったという。この処理で実行された命令数はいくつか。ただし、1ステップの実行時間を10n秒とし、すべての命令は途中で停止することなく実行されるものとする。

12 次の条件のとき、機械語命令でインデックス修飾をもつてオペランドを指定すると、実効アドレスはいくらになるか。

インデックスレジスタの値	10
命令語のアドレス部の値	300
命令が格納されているアドレス	2000

13 次の説明に最も関連しているものを解答群から選びなさい。

①一つの命令で同時に複数のデータの並列処理を行う。

②複数の命令で同一のデータに複数の異なる並列処理を行う。

③複数の命令で複数のデータに複数の異なる並列処理を行う。

④一つの命令で一つのデータ処理を行う。

解答群

SISD SIMD MISD MIMD

3. 3 主記憶装置

要 点

主記憶装置は実行するプログラムとデータを記憶するための装置で、信頼性、高速応答性、消費電力、経済性などで優れている半導体メモリが用いられる。



①キャッシュメモリ

キャッシュメモリは主記憶装置と制御装置の間にあって、主記憶装置のプログラムやデータの一部を一時的に蓄えておく、高速動作可能な記憶装置である。通常プログラム動作中は、一度読み出されたデータやプログラムが続けて読み出されることが多い。読み出された部分の前後をまとめて主記憶からキャッシュメモリにコピーし、キャッシュメモリをアクセスすることで高速な処理が可能となる。

②仮想記憶方式

主記憶装置は高速であるが容量が小さい。逆に外部記憶装置は大容量であるがアクセス速度は遅い。そこで、主記憶装置に入り切れないような大きなプログラムは、いくつかに分割して主記憶装置と補助記憶装置に分けて格納しておく、そして補助記憶装置にあるプログラムを実行するときは、主記憶装置の一部分と入れ替えながら(スワップという)実行する方法を仮想記憶方式という。これによって主記憶装置の見掛けの記憶容量を実際の容量よりも大きくできる利点がある。

③メモリの読み取りと書き込み

1. アクセス時間

主記憶装置が、制御装置から読み取りまたは書き込みの指令を受けてから、指示されたデータの受渡しが完了するまでの時間を、アクセス時間またはアクセスタイムといふ。

2. サイクル時間

同一のアドレスに対して読み取り(または書き込み)を続けて行ったとき、ある読み取り(または書き込み)から次の読み取り(または書き込み)ができるまでの時間をサイクル時間またはサイクルタイムといふ。通常サイクル時間はアクセス時間の2倍程度である。

□例題3.6□ 次の文中の [] に入るものを解答群から選びなさい。

- (1) 一般に、記憶装置への情報の書き込み、読み取りには、ある程度の [①] を要する。記憶装置の同一場所に連続して書き込み(または読み取り)を行う場合、ある一つの書き込み(または読み取り)が始まってから次の書き込み(または読み取り)が始まるまでに要する時間を [②] と呼ぶ。また、制御装置が情報を要求した瞬間から、情報の受渡しが完了する瞬間までに要する時間を [③] という。
- (2) 一般にMOS形メモリでは、蓄積された電荷が漏れて数ミリ秒後に記憶内容が失われる。そのため、この形のメモリでは記憶内容が失われないうちに再生する必要がある。それを [④] といい、そのようなメモリ素子を [⑤] 形という

解答群

- | | | | |
|------------|------------|-----------|------------|
| ア. サイクルタイム | イ. アキュムレータ | ウ. ROM | エ. 時間 |
| オ. リフレッシュ | カ. スタティック | キ. ダイナミック | ク. アクセスタイム |

解説

(1) 要点参照。

(2) ICメモリはMOS形とバイポーラ形に分類される、MOS形のダイナミックRAMは、非常に絶縁抵抗の大きいFETをコンデンサとして利用し、このコンデンサの電荷の有無によって情報を記憶する。しかし、電荷は時間とともに放電するので、放電し切らないうちに同じ内容を繰り返して書き込む再書き込み動作(リフレッシュ)が必要である。リフレッシュが必要なメモリをダイナミック形といい、不要なものをスタティック形という。

解答

- ①エ ②ア ③ク ④オ ⑤キ

□例題3.7□ 次の表の条件のときの実効メモリアクセス時間を求めなさい。

キャッシュメモリ		主記憶
アクセス時間	ヒット率	アクセス時間
10[n秒]	70[%]	60[n秒]

解説

キャッシュメモリのヒット率が70%のとき、主記憶のヒット率は $100 - 70 = 30\%$ となる。したがって、キャッシュメモリのアクセス時間を T_c 、主記憶のアクセス時間を T_m 、キャッシュメモリのヒット率を H 、実効メモリアクセス時間を T とすると、

$$T = T_c \times \left(\frac{H}{100} \right) + T_m \times \left(\frac{100-H}{100} \right)$$

と表せるので、

$$T = 10 \times 0.7 + 60 \times 0.3 = 25\text{n秒}$$
となる。

解答

25n秒

練習問題

14 次の文中の [] に入るべき語を解答群から選びなさい。(第27回)

- (1) メモリには読み書きできるRAMと読み取り専用のROMがあり、RAMには電源を切らない限り [①] 回路で記憶情報を保持する [②] RAMと、電荷の状態で情報を記憶するもので、数ミリ秒以内で電荷の大部分を失う [③] RAMの2種類がある。このため [③] RAMではリフレッシュという動作を行い、記憶情報を保持しなければならない。また、ROMでは、記憶情報を紫外線で消去したり、書き込み装置によって新たに書き込みが可能な [④] ROMや、電気的に消去と書き込みが可能で専用の書き込み器を必要としない [⑤] ROM、さらに工場で生産時にデータを書き込んでしまう [⑥] ROMなどがある。
- (2) 記憶装置に連続して書き込み(または読み取り)を行う場合、ある一つの書き込み(または読み取り)が始まってから、次の書き込み(または読み取り)が始まるまでに要する時間を [⑦] 時間という。
- (3) 記憶場所が0番地から順番に割り付けられている主記憶装置において、この主記憶装置をアクセスするアドレス長が16ビットであるとき、最大のアドレスは [⑧] 番地である。また、1つのアドレスに8ビット記憶するとき、この主記憶装置のメモリ容量は [⑨] kバイトである。
- (4) ハードウェアの基本的な動作を制御するためにROM化されたソフトウェアを [⑩] という。ソフトウェアとハードウェアの中間的な性格を持つので、このように呼ばれる。

解答群

- | | | | | | |
|-----------|-----------|-----------|-------|-----------|-------------|
| ア. リフレッシュ | イ. マスク | ウ. ソフトウェア | エ. EP | オ. EEPROM | カ. フリップフロップ |
| キ. スタティック | ク. フームウェア | ケ. 32 | コ. 64 | サ. ダイナミック | シ. アクセス |
| ス. サイクル | セ. 32767 | ソ. 65536 | | | |

15 次の説明に最も関連しているものを解答群から選びなさい。

- ①電気信号でデータの書換え・消去が可能で、電源を切っても内容を保持するメモリ。
- ②フリップフロップ回路によりデータを記憶するメモリ。
- ③数ミリ秒というような時間ごとにデータのリフレッシュが必要なメモリ。
- ④工場でデータを書き込むため、データの変更はできないメモリ。

解答群

DRAM	SRAM	フラッシュメモリ	マスクROM
------	------	----------	--------

16 キャッシュメモリのアクセス時間10n秒、主記憶のアクセス時間が50n秒のとき、キャッシュメモリのヒット率が70%であるとすると、実効アクセス時間は何n秒であるか。

17 次の説明に最も関連しているものを解答群から選びなさい。

- ①キャッシュメモリと主記憶に同時にデータを書き込む。
- ②主記憶を幾つかの並列にアクセスできる区画に分割しアクセスを高速化する。
- ③キャッシュメモリにだけデータを書き込み、キャッシュメモリからデータを追い出すときに主記憶へもデータを書き込む。

解答群

FIFO	LIFO	ライトバック	ライトスルー	メモリインターブ
------	------	--------	--------	----------

18 次の説明に最も関連しているものを解答群から選びなさい。

- ①最近参照されず、参照されてから最も時間が経過しているデータを対象にする。
- ②先入れ先出しあともいい、最初に参照されたデータを対象にする。
- ③後入れ先出しあともいい、最後に参照されたデータを対象にする。
- ④参照回数が少なく、使用頻度の少ないデータを対象にする。

解答群

FIFO	LFU	LIFO	LRU
------	-----	------	-----

19 次の文中の [] に入るべきものを解答群から選びなさい。

- (1) 主記憶装置のある場所を指定し、最後に記憶したものを最初に取り出す [①] の形でデータを記憶するものを [②] といい、サブルーチン関係の命令やデータ待避命令などに使われる。
- (2) [③] アドレス方式はアドレスの指定を、その命令の所在場所から±nの形で指定する。この方式は、[④] の中で、プログラムを自由に配置し得る [⑤] なプログラムとすることができる。命令語で指定した場所に、その命令の作用する [⑥] が入っている方式を [⑦] アドレス指定方式といい、他のプログラムと [⑧] 等を行う際に必要なアドレス指定方式である。
- (3) アセンブラー語には、一つまたは複数個の機械語に変換される [⑨] 命令と、記憶場所や実行開始アドレスなどの指示をする [⑩] 命令がある。後者は対応する機械語がないから、機械語には変換されない。

解答群

ア. 主記憶装置	イ. 擬似	ウ. 結合・編集	エ. アドレスの値	オ. 実行可能	カ. 再配置可能
キ. プッシュダウンスタック	ク. ラストイン・ファーストアウト	ケ. 相対	コ. 間接		

20 次のDRAMとSRAMの相対的な特徴を比較した表に、入るべきものを解答群から選びなさい。

	集積度	速度	消費電力	製造コスト	用途
DRAM	①	③	⑤	⑦	⑨
SRAM	②	④	⑥	⑧	⑩

解答群

ア. 高	イ. 低	ウ. キャッシュメモリ	エ. 主記憶	オ. 補助記憶
------	------	-------------	--------	---------

21 次の文中の [] に入るべきものを解答群から選び記号で答えなさい。

- (1) [①]と主記憶の間にキャッシュメモリを置くことがある。通常、プログラムの動作中は、一度読み出されたデータやプログラムが続けて読み出されることが多い。読み出された前後の部分をまとめて主記憶からキャッシュメモリに [②] し、そのデータやプログラムを [③] することで [④] な処理が可能となる。
- (2) 一般的に主記憶装置は高速であるが [⑤] が小さい。逆に外部記憶装置は [⑥] であるがアクセス速度は遅い。主記憶装置に入りきれない大きなプログラムを [⑦] 装置を利用して、あたかも主記憶装置の一部であるかのようにして使用する方法を [⑧] 方式と呼ぶ。

解答群

- | | | | | | | | |
|--------|-------|---------|----------|--------|---------|-----------|-------|
| ア. 高速 | イ. 演算 | ウ. 容量 | エ. RAM | オ. ROM | カ. 補助記憶 | キ. 主記憶 | ク. 低速 |
| ケ. コピー | コ. 補助 | サ. アクセス | シ. キャッシュ | ス. 大容量 | セ. 仮想記憶 | ソ. 中央処理装置 | |

3. 4 補助記憶装置

要 点

主記憶装置は半導体メモリで高速に応答することが可能であるが、不揮発性で大容量のものは高価になるため、多数のプログラムやデータなどのファイルは、磁気ディスクなどアクセス時間は多少かかるが大容量である補助記憶装置に格納される

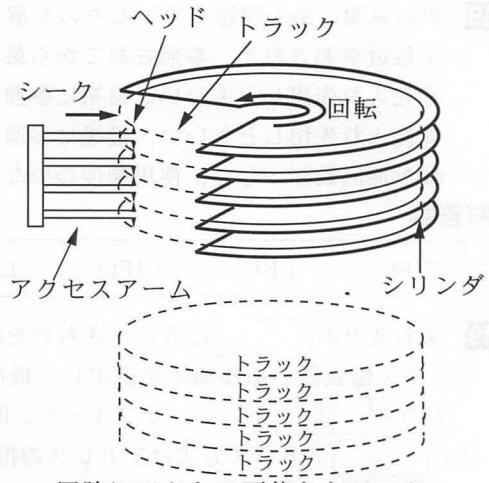
① 磁気ディスク装置

磁気ディスク装置は、両面に磁性材料を均一に塗布したアルミニウムなどの円盤を一定速度で回転させ、各面を水平に移動する磁気ヘッドによってデータの書き込み・読み取りを行う装置である。磁気ヘッドが目的のトラックに移動することをシーク動作といい、これに要する時間をシーク時間(シークタイム)という、また、シーク動作の後、読み取り・書き込みの場所が磁気ヘッドの真下にくるまでの時間を回転待ち時間(サーチ時間)といい、最長1回転に要する時間である。この半分の時間をとって平均回転待ち時間という。アクセス時間は両者の和にデータ転送時間を加えたものになる。

アクセス時間 = シーク時間 + 回転待ち時間 + データ転送時間
このとき、磁気ヘッドの付いたアクセスアームはくし状に並んでいるため、アクセス可能なトラックがディスク面の数だけ円筒状(シリンドラ状)に並ぶことになりこれをシリンドラと呼ぶ。

② RAID(Redundant Array of Inexpensive Disks)

ハードディスクを複数台並列に並べ、それら全体を1つのディスク装置のように制御することにより、アクセス速度を向上させたり、トラブルに対する耐久性や信頼性をあげることができる。このような装置をディスクアレイといい、低価格ディスクを組み合わせて、高性能なディスク装置として利用できる。



同時にアクセス可能なトラックを仮想的な筒とし、シリンドラと呼ぶ

□例題3.8□

28ビットで表現されたセクタ番号で、1セクタ512バイトの磁気ディスク装置を管理するとき、最大何Gバイトまで管理できるか。ただし、1Gバイト = 1×2^{30} バイトとする。

解説

$$(512 \times 2^{28}) \div (1 \times 2^{30}) = (2^9 \times 2^{28}) \div (1 \times 2^{30}) = 2^{37} \div 2^{30} = 2^7 = 128$$

解答

128Gバイト

□例題3.9□

磁気ディスク装置に関する次の記述を読んで、設問中の [] に入るべき最も適切な数値を答えなさい。

(第30回)

平均シーク時間	8ミリ秒
回転速度	7500回転/分
トラック当りの最大記憶容量	187500バイト/トラック
ブロック間隔	150バイト
トラック/シリンドラ	8トラック
シリンドラ/ディスク	30000シリンドラ

- (1) このディスク装置は、ディスク当りの記憶容量から [①] Gバイトのディスクと呼ばれる。
(ただし、1Gバイト=1 000 000 000バイトとする。)
- (2) この磁気ディスク装置に、600バイト長のレコード15万件を1レコード/ブロックで記録するとき、最低 [②] シリング必要である。
- (3) 上記と同じ600バイト長のレコード15万件を、10レコード/ブロックで記録するとき、最低 [③] シリンダ必要である。
- (4) この磁気ディスクの平均回転待ち時間は、[④] ミリ秒である。
- (5) この磁気ディスク装置に、75000バイト長のレコードが記録されているとき、このレコードの転送時間は [⑤] ミリ秒であるので、呼出し時間(平均シーク時間、平均回転待ち時間とデータ転送時間の和)は [⑥] ミリ秒である。

解答群

ア. 1.6	イ. 2	ウ. 3.2	エ. 4	オ. 8	カ. 9.6	キ. 15.2	ク. 18	ケ. 18.6	コ. 19
サ. 29	シ. 30	ス. 31	セ. 32.6	ソ. 40	タ. 41	チ. 45	ツ. 63	テ. 64	ト. 75
ナ. 86	ニ. 92	ヌ. 152							

解説

- (1) 磁気ディスクの容量はトラック当りの容量×シリンドラ当りのトラック数×ディスク当りのシリンドラ数から求まるので、数値を代入して、 $187\ 500 \times 8 \times 30\ 000 = 45\ 000\ 000\ 000 = 45\text{Gバイト}$
- (2) 1レコードを600バイト、ブロック間隔を150バイトとすると、1ブロック750バイトで構成される。
トラック当りのブロック数は、 $187\ 500 \div 750 = 250$ ブロック
全データを記録するのに必要なトラック数は、 $150\ 000 \div 250 = 600$ トラック
シリンドラ当りのトラック数は8であるから、必要なシリンドラ数は、 $600 \div 8 = 75$ シリンドラ
- (3) 10レコード/ブロックの場合は、1ブロック当り $600 \times 10 + 150 = 6150$ バイト長となる。
トラック当りのブロック数は、 $187\ 500 \div 6150 = 30.48\cdots$ 、30ブロック
全データを記録するのに必要なトラック数は、 $150\ 000 \div 10 \div 30 = 500$ トラック
したがって、シリンドラ数は、 $500 \div 8 = 62.5$ 、63シリンドラ
- (4) 1回転の時間= $1 \div (7500 \div 60) = 0.008 = 8$ ミリ秒
平均回転待ち時間=1回転の時間÷2
 $= 8 \div 2 = 4$ ミリ秒
- (5) 平均シーク時間=8ミリ秒
データ転送時間=データ長/転送速度
ここで転送速度は、1トラックの容量を1回転で送る速さに等しいので
 $187\ 500 \div 0.008 = 23\ 437\ 500$ バイト/秒
転送時間は、 $75000 \div 23\ 437\ 500 = 0.0032 = 3.2$ ミリ秒
以上から、呼出し時間= $8 + 4 + 3.2 = 15.2$ ミリ秒を得る。

解答

- ①チ ②ト ③ツ ④エ ⑤ウ ⑥キ

練習問題

22 次の説明に最も適切な語を解答群から選び、記号で答えなさい。(第34回)

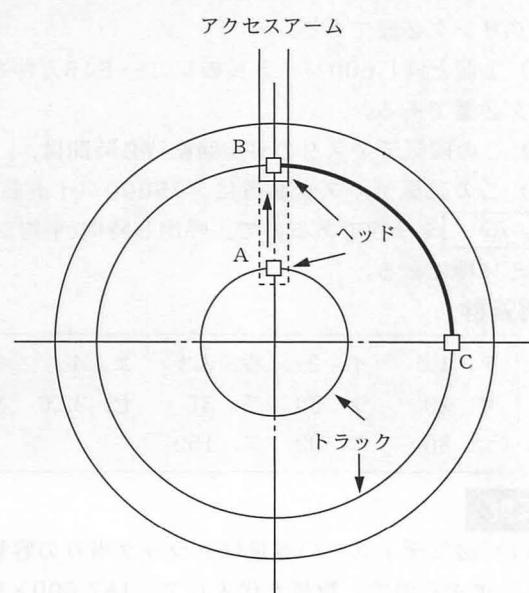
パソコンなどで使用される記憶媒体には、情報を読み出すだけのものや、読み書きすることが可能なものが
ある。レーザ光を用いてプラスチック盤にプレスされたピットとランドを読みとるDVD-①は、情報を
読み出すだけの記憶媒体である。レーザ光を用いて記録層を結晶状態または非結晶(アモルファス)状態に変化
させて情報を記録する方式を用いるDVD-②では、情報を書き換えることが可能である。また、情報の
書き込みはできるが書き換えのできないDVD-③は有機色素をレーザー光で焼き付けて情報を記録して
いる。また、磁気で情報を記憶する④や、記録面にレーザ光を照射し磁気と熱で磁化の方向を変えること
でデータを記録する⑤は情報を書き換えることができる記憶媒体である。

解答群

- | | | | | | | |
|------|-------|---------|-------|---------|-------|--------|
| ア. R | イ. MO | ウ. SRAM | エ. RW | オ. PROM | カ. FD | キ. ROM |
|------|-------|---------|-------|---------|-------|--------|

23 次の説明に最も適切な語を解答群から選び、記号で答え
なさい。(第34回)

図は磁気ディスク装置の構造を示すものである。現在、
磁気ヘッドの位置がAにあり、Cの位置のデータを読み取る
場合を考えたとき、磁気ヘッドを取り付けたアームを動か
して、目的のトラック位置Bまで移動する。この動作を
①または位置決めといふ。Bに位置付けられた磁気ヘ
ッドは磁気ディスクの回転により、Cの位置の目的データが、
Bの磁気ヘッドの下にくるのを待つ。これを②または
回転待ちといふ。一般に、平均回転待ち時間は、磁気ディ
スクが1回転する時間の③倍となる。磁気ディスク
装置のアクセス時間は平均①時間と、平均回転待ち
時間、そして④時間の和になる。④時間は、
アクセスするデータ長と磁気ディスクの⑤、1トラ
ックあたりの記憶容量によって決定される。



解答群

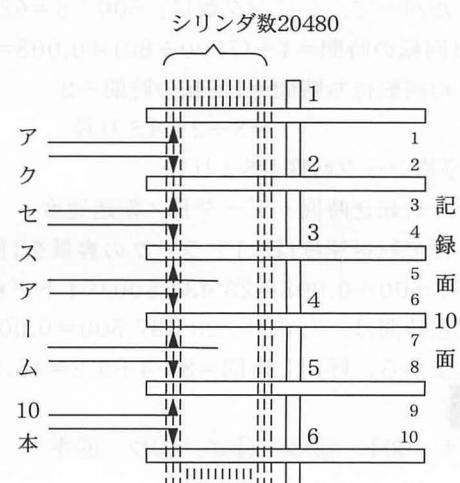
- | | | | | | | | |
|---------|--------|---------|---------|--------|----------|---------|--------|
| ア. 1 | イ. 2 | ウ. 0.5 | エ. 0.25 | オ. 書込み | カ. データ転送 | キ. 回転速度 | ク. サーチ |
| ケ. ブロック | コ. シーク | サ. サイクル | シ. 実行 | | | | |

24 次の仕様でフォーマットしたフロッピーディスク1枚分の記憶容量は何キロバイトか。小数第1位まで求め
なさい。ただし、1キロバイトは1024バイトとする。(第31回)

総トラック数	160
セクタ数 トラック数	9
セクタ長	512バイト

25 図に示す磁気ディスク記憶装置において、シリンドラ数が
20480、1シリンドラ当たりのトラック数10、1トラック当り
の記憶容量を256kバイトとすると、この記憶容量は何Gバ
イトとなるか。

ただし、1Gバイト=1024Mバイト、1Mバイト=
1024kバイトとする。(第31回)



- 26 次の文をよく読んで、 RAIDに関係のない説明を選び記号で答えなさい。(第28回)
- ア.複数のディスクにデータを特定サイズごとに書き込み、特定の装置にアクセスを集中させず効率をよくすることを、ストライピングという。
- イ.2台のディスクに同じデータを記録し、安全性を高めることを、ミラーリングという。
- ウ.低速な磁気ディスクと主記憶装置の間におき緩衝記憶の役割を果たすものを、ディスクキャッシュという。
- エ.複数のディスクアレイに対して並行的にアクセスするので高速である。

- 27 次の文中の [] を埋めなさい。(第27回)

表の仕様の磁気ディスク装置において、1セクタ分のデータの呼び出しが終了するまでの平均時間を求めたい。まず、回転数からこのディスクが1回転するのに約 [①] ミリ秒必要であるのがわかる。これから、平均回転待ち時間は約 [②] ミリ秒と計算できる。1セクタのデータ長は512バイトであるので、1kバイトを 10^3 バイトとすると、1セクタ分のデータ転送時間は約 [③] ミリ秒となる。したがって、1セクタ分だけのデータ呼出しに必要な平均時間は約 [④] ミリ秒となる。

データ長	512バイト／セクタ
回転数	6000回転／分
平均シーク時間	7ミリ秒
データ転送速度	128kバイト／秒

- 28 次の記述のうち適切でないものはどれか。

- ア.論理フォーマットでは、OSのファイルシステムの管理領域やディレクトリ領域、クラスタなど、記録するデータの論理的な位置を設定する。
- イ.物理フォーマットでは、セクタやトラックなど物理的な区画を作成して物理的なアドレスを割り振る。
- ウ.論理フォーマットでは、不良セクタの検査も行い、不良セクタが発見されたときには予備領域内の正常なセクタを不良セクタの代替とする処理を行う。
- エ.磁気ディスクでは、物理フォーマットをおこなってから論理フォーマットを行う。

- 29 パソコンの磁気ディスクを長期間にわたって使用しているうちに、ファイルのフラグメンテーション(断片化)が大量に発生して、処理速度が遅くなったように感じた。次の記述のうち適切なものはどれか。

- ア.あるファイルにフラグメンテーションが発生すると、フラグメンテーションが発生する前の元のファイルよりサイズが大きくなる。
- イ.フラグメンテーションを解消するにはデフラグを行うとよい。
- ウ.フラグメンテーションが発生したファイルを移動すると、移動先でフラグメンテーションが更に進行し、フラグメンテーションが解消することはない。
- エ.磁気ディスク全体を論理フォーマットすればフラグメンテーションは解消する。

- 30 CD-Rの書き込み方式に関する次の説明に最も関連しているものを解答群から選びなさい。

- ①データを小さな単位で書き込むことができるので、磁気ディスクに書き込むのと同じような感覚でデータを書き込める。
- ②一つのセッションからなるため、CD-ROMと同じように追記することができない。
- ③複数のセッションからなり、追記が可能である。

解答群

ディスクアットワنس	トラックアットワنس	パケットライティング
------------	------------	------------

- 31 磁気ディスク装置の入出力要求の待ち行列にシリンド番号130, 70, 150, 50, 200, 80, 90が並んでいる。現在、磁気ディスク装置のヘッドがシリンド番号100にあるとすると、ヘッドが移動するシリンドの総数はいくらか。ただし、待ち行列に並んでいる入出力要求を並べ替えて、できるだけヘッドを一方向に動かしシリンド番号順に処理するシーク最適化方式によるものとする。また、現在、ヘッドはシリンド番号が増加する方向に移動しているため、次は増加する方向への移動から開始するものとし、移動方向に要求がないとき反転するものとする。他の要因が結果に影響することはない。

32 磁気ディスクについて次の各問の答を解答群から選びなさい。ただし、磁気ディスクは6枚10面構成で、10個のヘッドが平行移動するタイプである。ただし、1Mバイト=1000000バイトとする。

- ①このディスクの、シリンドラ当たりのトラック数はいくつか。
- ②シリンドラ数を200とするとき、ディスク全体ではいくつのトラックが存在することになるか。
- ③1トラックに12000バイトのデータを記録することができるとき、ディスク全体の記憶容量は何Mバイトか。
- ④このディスクの回転数が3000回転／分であるとき、平均回転待ち時間は何ミリ秒か。
- ⑤1トラックに12000バイト記録してあるデータを読むときのデータ転送速度は何kバイト／秒か。

[①と③に関する解答群] ア. 6 イ. 8 ウ. 10 エ. 24

[②に関する解答群] ア. 200 イ. 400 ウ. 800 エ. 1000 オ. 2000

[④に関する解答群] ア. 10 イ. 20 ウ. 30 エ. 40 オ. 50

[⑤に関する解答群] ア. 120 イ. 240 ウ. 360 エ. 480 オ. 600

33 次の各問に対する答をすぐ下の解答群から選びなさい。

- ①パソコンなどで利用される記憶媒体を、記憶容量の大きい順に並べなさい。

ア.FD, CD-ROM, DVD-ROM イ.DVD-ROM, CD-ROM, FD

ウ.CD-ROM, DVD-ROM, FD エ.CD-ROM, FD, DVD-ROM

- ②MOディスクに関する説明で正しいものを選びなさい。

ア.CD-ROMと同じようにレーザー光だけでデータを読み書きする。

イ.FDより記憶容量が多いのでアクセス速度も遅い。

ウ.一般的に3.5インチのMOディスクの容量は、DVD-ROMの容量より大きい。

エ.記録膜にレーザー光を照射し、磁気と熱で磁化の方向を変えることでデータを記録する。

- ③RAIDの説明のうち正しくないもの選びなさい。

ア.複数のディスクにデータを特定サイズごとに書き込み、特定の装置にアクセスを集中させず効率をよくする。

イ.2台のディスクに同じデータを記録し、安全性を高める。

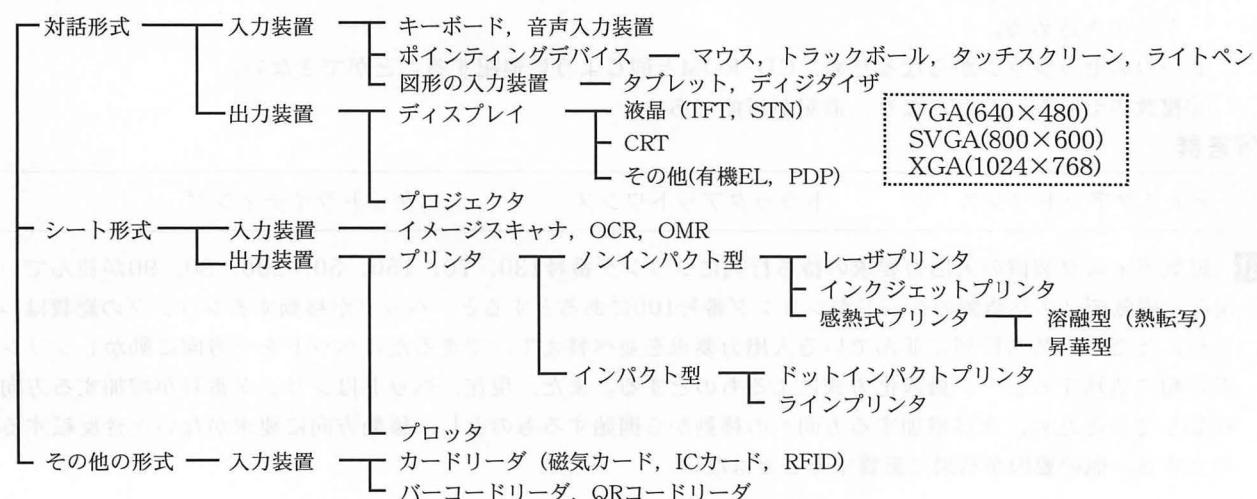
ウ.低速な磁気ディスクと主記憶装置の間におき緩衝記憶の役割を果たす。

エ.複数のディスクアレイに対して並行的にアクセスするので高速である。

3. 5 入出力装置

要 点

〔1〕入出力装置の種類



ディスプレイやキーボードを用いてコンピュータと情報を交換しながら処理を進める方法を対話形式という。一方、カードや紙などのシートを用いて情報の入出力を行う方法をシート形式という。また、キーボードとディスプレイを備えた入出力端末をVDT(Video Display Terminal)と呼び、ほとんどのコンピュータ端末やネットワークに接続されたワークステーションやパソコンを指し、対話型の代表的な存在である。

[2] 入出力インターフェース

入出力装置とコンピュータを接続するためには、コネクタの形状や電気的条件、データの転送方法などの取り決めに従う必要がある。これらの取り決めをインターフェースという。

インターフェース	接続機器	転送方式
USB	マウス、キーボード、ハードディスク等	シリアル
RS-232C	モデム	シリアル
SCSI	ハードディスク、イメージスキャナ等	パラレル
セントロニクス	プリンタ	パラレル
GPIB	計測機器	パラレル
IEEE-1394	デジタルビデオ、ハードディスク等	シリアル
IDE	ハードディスク等	パラレル
IrDA	ノートパソコンとPDAの通信等	シリアル

[3] 入出力制御

高速に処理できる処理装置に比べると、補助記憶装置やプリンタなどの周辺装置ははるかに低速である。このため、主記憶装置と周辺装置との間に入出力チャネルと呼ばれる専用の制御装置を入れ、コンピュータ全体の処理能力の向上を図っている。演算処理と同時に複数の入出力処理を独立に行うことができ、いわゆる並列処理が可能となる。

□例題3.10□

次の説明に最も適切な語を解答群から選びなさい。(第30回)

- ①パソコンが普及していた初期の頃よりシリアルインターフェースとして利用され、モデムなどが接続された。
COMポートと呼ばれることもある。
- ②記録膜にレーザー光を照射し、磁気と熱で磁化の方向を変えることでデータを記録する。
- ③スカジーと読み、パソコンとハードディスク、MO、CD-R、イメージスキャナなどの高速な周辺装置とのインターフェース。
- ④パソコンに主としてプリンタを接続するためのパラレルインターフェース。
- ⑤IEEE-488規格ともいい、パソコンとマイコン内蔵計測機器などの周辺装置とのインターフェース。
- ⑥従来のシリアル接続に比べるとはるかに高速なシリアル通信が可能なインターフェースで、マウスやキーボードの他に、ハードディスク、MO、CD-R、イメージスキャナなどの高速な周辺装置とのデータ転送が可能である。一つのポートで、ハブを用いることにより複数の機器を接続できる。また、電源を切らずに接続の抜き差しができるなどの特徴を持つ。
- ⑦データ媒体に手書きされたマークを光学的に検出し、コンピュータに入力する装置。
- ⑧i.LINK、FireWireとも呼ばれ、デジタルビデオなどの高速なデータ転送が可能なシリアルインターフェースで、ハードディスクなどのデータ転送にも使用される。ホストとなるパーソナルコンピュータを必要とせず、機器同士の通信も可能になっている。
- ⑨パソコンと内蔵用のハードディスクなどを接続するための標準的なパラレルインターフェースで、1本のケーブルでマスター、スレーブの2台の接続ができる。最近のパソコンでは、プライマリとセカンダリの2つのケーブル接続口があり、合計4台のハードディスクやCD-ROMドライブなどを接続できることが多い。
- ⑩赤外線を用いてノートパソコンとプリンタなど近距離での通信が可能である。

解答群

- | | | | | | | |
|------------|------------|--------------|--------|---------|---------|---------|
| ア. OCR | イ. IrDA | ウ. IEEE-1394 | エ. USB | オ. CD-R | カ. GPIB | キ. SCSI |
| ク. RS232-C | ケ. セントロニクス | コ. SASI | サ. MO | シ. OMR | ス. IDE | |

解説

MO(Magneto Optical), SCSI(Small Computer System Interface), GPIB(General Purpose Interface Bus)
USB(Universal Serial Bus), OMR(Optical Mark Reader),
IEEE-1394(the Institute of Electrical and Electronics Engineers 1394),
IDE(Integrated Drive Electronics), IrDA(Infrared Data Association)

解答

- ①ク ②サ ③キ ④ケ ⑤カ ⑥エ ⑦シ ⑧ウ ⑨ス ⑩イ

□例題3.11□

1画面を横1024ドット、縦768ドットで表示するカラーディスプレイがある。1ドットにつき、赤、緑、青(RGB)の三原色それぞれ8ビットの色情報量を持っているとすると、1画面分の情報を記憶するために必要なビデオメモリの容量は約何Mバイト必要か。小数第2位を切り上げて求めなさい。ただし、1Mバイト=1024kバイト、1kバイト=1024バイトとする。(第32回)

解説

1ドットにつき、RGBの3色で24ビット(3バイト)の色情報を持っている。

1画面分の情報量は、 $3 \times 1024 \times 768 = 2,359,296$ バイト

$2,359,296$ バイト $\div (1024 \times 1024) = 2.25 \approx 2.3$ Mバイト

となる。

この問題では、次のように式を書くと計算が簡単になる。

$$\begin{array}{r} 3 \times 1024 \times 768 \ 3 \\ \hline 1024 \times 1024 \ 4 \end{array}$$

コンピュータは2進数を基準に設計されているため、 $2^1, 2^2, 2^3 \dots$ が因数になっている場合が多いので、工夫すると簡単に計算できる場合がある。

計算の早い段階で8ビットを1バイトに換算したり、画面の横1024ドットと1Mバイト=1024kバイトのように問題文から簡単な計算方法を考えることも大切である。

解答 2.3Mバイト

練習問題

34 イメージスキャナを使って、縦横それぞれ600[dpi]の品質で縦10cm×横5cmの写真を24ビットカラーで読み込んだときのデータは約何Mバイトか。小数第2位を四捨五入して求めなさい。ただし、[dpi]は、1インチあたりの画素数(ドット数)の単位とし、24ビットカラーでは読み込んだそれぞれの画素(1ドット)につき24ビットのデータが必要であるとする。また、1インチ=2.5cm、1kバイト=1000バイト、1Mバイト=1000kバイトとする。(第33回)

35 パソコンのモニターに、横720ドット×縦480ドットで256色の動画を表示させたい。この動画が30フレーム/秒であるときこのとき、1分間に表示される画像データの量は約何Mバイトか。ただし、1Mバイト=1×10⁶バイトとし、データは圧縮しないものとする。

36 次の説明に最も関連しているものを解答群から選びなさい。

- ①カード表面に磁性体を持ち磁気記録する。
- ②集積回路をカード内部に内蔵しデータの記憶や演算を行う。
- ③集積回路とアンテナをカード内部に内蔵し、非接触でデータの読み書きができる。
- ④二次元の模様でデータを記憶する。

解答群

QRコード

RFID

ICカード

磁気カード

37 次の各間に答えなさい。

- ①1画面を800×600ドットで表示するカラーディスプレイがある。1ドットにつき、赤、緑、青(RGB)の三原色それぞれ8ビットの色情報量を持っているとすると、1画面分の情報を記憶するために必要なビデオメモリの容量は約何Mバイト必要か。ただし、1Mバイト=1 000 000バイトとし、小数第2位を四捨五入して求めなさい。
- ②1ドット分の色情報が24ビットのとき表現できる最大の色数は約何万色か。小数第1位を四捨五入して求めなさい。
- ③音声を1秒間に22,000回等間隔にサンプリングし、16ビットのモノラルデータとして記録するとき、1.44Mバイトの容量のフロッピーディスクには約何秒のデータを記録できるか。ただし、1Mバイト=1 000 000バイトとし、サンプリングデータのみを記録するものとする。

38 次の各問に対する答をすぐ下の解答群から選びなさい。

- ①次のインターフェースのうちデータを直列伝送するものを選びなさい。

解答群

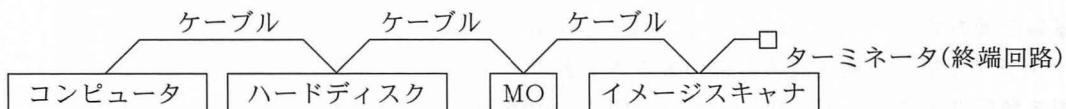
ア. SCSI イ. セントロニクス ウ. GPIB エ. USB

- ②次のディスプレイのうち、表示速度の速い順に並べなさい。

解答群

ア. TFT, D-STN, STN イ. STN, D-STN, TFT
ウ. D-STN, STN, TFT エ. TFT, STN, D-STN

- ③各周辺装置を図のような接続で使用することのできるインターフェースを選びなさい。



解答群

ア. SCSI イ. RS-232C ウ. IrDA エ. USB

39 入力装置に関する次の記述①～⑤にそれぞれ最も関連の深い字句を解答群から選びなさい。

- ①ディスプレイ装置の蛍光面の表示箇所を指示すると、光電素子を利用してその位置を検出し、コンピュータに入力する装置。
- ②人間がトレースする平板上の点の座標を検出し、コンピュータに入力する装置。
- ③平らな面上で動かすことにより、移動方向と移動量を検出し、コンピュータに入力する装置。手の中にに入るくらいの大きさで、1～3個のボタンが付いている。
- ④太さや間隔の異なる縦線の組合せによってコードを読み取り、コンピュータに入力する装置、POS端末によく利用される。
- ⑤データ媒体に手書きされたマークを光学的に検出し、コンピュータに入力する装置。

解答群

ア. OMR イ. OCR ウ. プロッタ エ. ジョイスティック
オ. ディジタイザ(タブレット) カ. バーコードリーダ キ. マウス ク. ライトペン

4. 通信

4. 1 通信方式

要 点

[1] 伝送方式

①直列伝送と並列伝送

1. 直列伝送方式(シリアル伝送方式)

複数のデータを1ビットずつ伝送する方式である。8ビットのデータを伝送するときには1本の通信線で1ビットずつ8回に分けて伝送する。長距離の伝送に適している。

2. 並列伝送方式(パラレル伝送方式)

複数のデータを複数の伝送路で同時に伝送する方式である。8ビットのデータを伝送するときには8本の通信線を用いる。伝送速度が速い。

②通信方式

1. 単方向通信

ある端末から他の端末への一方向のみの伝送が可能である。

2. 半二重通信

物理的には一方向伝送であるが、片側通行のように送受を切り替えて双方向の伝送を行う。

3. 全二重通信

同時に双方の伝送が可能である。

③変調方式

1. 振幅変調方式 (ASK : amplitude shift keying)

ディジタル信号を搬送波の振幅の大きさで表す。

2. 周波数変調方式 (FSK : frequency shift keying)

ディジタル信号を搬送波の周波数の高さで表す。

3. 位相変調方式 (PSK : phase shift keying)

ディジタル信号を搬送波の基準位相との位相差で表す。

④通信速度

1. 変調速度

1秒間に変調された回数で、単位にはbaud(ボー)を使用する。

位相などの変化が1秒間に何回あるかが、変調速度に相当する。

2. データ信号速度

1秒間に伝送できるビット数で、単位には、bit/s(bps)を使用する。

3. データ転送速度

文字/秒、文字/分、ブロック/分など単位時間に伝送できるデータ量の平均値である。

⑤同期

データ通信を行うとき、送信側と受信側とが同じ形式の通信規約にのっとり、双方がタイミングをとりながら送受をする必要がある。このタイミングをとることを同期という。同期には、大きく分けて、ビット同期と、ブロック同期がある。

1. ビット同期方式

1ビットごとに同期をとる方式である。

2. ブロック同期方式

数ビットのブロックごとに同期をとる方式である。

(a) 調歩同期方式

8ビットの文字符号の前にスタートビット、後にストップビットを付加して同期をとる方式で、8ビットのデータ伝送のためには合計10ビットが必要となる。非同期式ともいわれる。



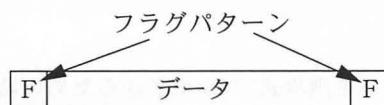
(b) キャラクタ同期方式

SYN符号(00010110など)を全データの先頭に付加して同期をとる方式である。



(c) フラグ同期方式

データの前後に必ずフラグパターン01111110を送り続けて同期をとる方式である。データが送信されていないときにも、フラグパターンが流れ続けている。



3. プロトコル

データ伝送の際の通信に関する約束事で、通信規約ともいう。インターネットでは、UNIXでの使用実績によりTCP/IP(Transmission Control Protocol / Internet Protocol)が採用され、事実上の標準となっている。

⑥ 伝送制御手順

1. ベーシック手順

送信側と受信側で伝送制御キャラクタをやりとりして、通信状況を確認しながらデータを伝送する方法。

2. HDLC手順(High level Data Link Control procedure)

データを制御情報などと一緒にフレームにして伝送する方法。



[2] 通信回線

① 通信回線

1. 専用回線(特定通信回線)

利用者が通信会社との契約によって固定的に設置する専用の回線。

2. 交換網の使用(公衆通信回線)

特に固定的な回線は設定せず、だれでも必要なときに利用できる回線網。

② 通信網

1. アナログ通信網

電話回線と同じアナログ通信網を利用するデータ通信で、ディジタル符号をアナログ回線で伝送するために、端末と回線との間にモデム(modem, 変・復調器)が必要である。一般的な電話回線に接続するモデムは、最大56kbit/sのものが多い。

2. ディジタル通信網

ディジタル信号のまま伝送し、交換できる通信回線網で、高速・高品質のデータ通信が行える。これには、時分割多重の技術を利用した回線交換方式と、蓄積交換の技術を利用したパケット交換方式がある。

(a) パケット交換網

データをパケット(小包)と呼ばれるパック形式(例256バイト)に分割して伝送する。各パケットには相手のコード(ヘッダ情報)が入っており、交換網はパケットごとに相手をチェックしながら蓄積し、空いている回線を探して順次転送するので回線網を効率よく利用でき、物理的に1本の回線でも多重化が可能となる。端末相互間でデータをやりとりするための手順(プロトコル)が必要となるが、交換器で制御手順や通信速度の変換が可能であり、交換器間などの各伝送段階でデータの誤りチェックがおこなわれるので、高品質の伝送が可能である。データ量(パケット量)に対して課金され、接続時間は長いがデータ量の少ない通信に適している。

(b) 回線交換網

普通の電話回線のように、データ伝送の必要が生じたときに端末と端末を電話回線や交換器を通して直接、電気的に回線を設定し、そのときだけディジタルデータ交換網の回線を占有する。回線の接続時間に対して課金され、接続時間は短いがデータ量の多い通信に適している。

□例題4.1□

9600bpsの通信回線を使って、8ビットの文字コードを調歩式で4800字送るための時間はどれだけか。ただし、スタートビット、ストップビットは各1ビットとする。

解説

直列にデータを伝送する場合、送信側と受信側でタイミングを合わせながらやり取りするために同期をとる必要がある。同期のとり方には調歩(非同期)式と同期式がある。同期式が、一定時間間隔で伝送するのに対して調歩式は送信データの前後に同期をとるためにスタートビットとストップビットを附加した符号の構成とし、この符号ごとに同期をとりながら送受信する。

8ビットの文字コードの場合は、前後に1ビットずつスタートビットとストップビットを附加すると10ビットの符号となり、4800文字では48000ビットを送ることになる。したがって、伝送速度を9600bpsとすれば、伝送時間は5秒ということになる。

解答 5秒

□例題4.2□

次の文は何について説明したものか、その名称を答えなさい。

- ①データ通信を行うときに、送信側と受信側があらかじめ決めておかなければならぬ通信規約。
- ②通信回線において、両方向の伝送はできるが同時に行えない回線。
- ③端末とアナログ回線の間に接続され、データ伝送の際、ディジタル信号をアナログ信号に変換して送出し、受信時には逆の変換をする装置。
- ④構内、建物など限られた場所に設置した汎用コンピュータやミニコンピュータ、パソコン等を高速伝送路で相互接続し、ファイルの共有化や分散処理を図るようにした通信網。

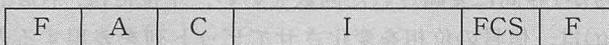
解説

- ①プロトコルといい、データ伝送の際の伝送速度やスタートビット、ストップビットの数、パリティビットの附加、送受信の手順など、通信に関する細かな約束を取り決めたもの。
- ②同時に双方向の通信はできないが、片方ずつならば通信が可能な方式を半二重回線といいう。
- ③モデムといい、変調と復調を行う装置。
- ④LANといい、local area networkの略称。

解答

- ①プロトコル ②半二重回線 ③モデム ④LAN

□例題4.3□ 図はHDLC手順のフレームの構成を表している。次の各間に答えなさい。



- (1) ①図において、アドレスを表しているフィールドはどれか。
②図において、誤り検出に用いられるフィールドはどれか。
③Fのビット列をいいなさい。
- (2) HDLC手順では、データの中にFと同じビット列があらわれた場合に区別する工夫がされている。受信側で0000 0101 1111 0100というビット列を受信したとき、送信側で送りたい元のデータはどのようなビット列であるか。

解説

- (1) 要点参照
(2) ビットスタッフィングに関する設問である。データがフラグF(01111110)と重複するビット列が存在する場合、一部のビットを変更して重複を避けている。
送信側では、1が5個続いた場合は、その後6個連續にならないように0を余分に挿入する。受信側では、1が5個続いた場合は、その後の余分な0を削除する。
設問では、1が5個連續している部分があるので、その後の0を削除すればよい。

0000 0101 1111 0100

ビットスタッフィングは、送信側ではFCS計算の後、受信側ではFCS計算の前におこなわれる。

解答

- (1) ①A ②FCS ③01111110
(2) 0000 0101 1111 100

練習問題

1 次の文章を読んで、下の各間に答えなさい。(第34回)

データ信号速度が14.4M[bps]の通信回線でデジタルカメラで撮影した画像データを伝送したい。ただし、[bps]とは、1秒間に伝送可能なビット数である。また、1Mbps=1000000bpsとする。

- (1) デジタルカメラで撮影した画像は、横1200画素、縦1000画素で、1画素を24ビットで表しているとすると、画像1枚のデータは何kバイトか求めなさい。ただし、画像データに画素以外のデータは含まれず、圧縮せずに保存する。また、1kバイト=1000バイトとする。
- (2) (1)の画像データを平均して20分の1に圧縮したデータとして伝送するとき、1秒間に何枚の画像データを伝送できるか求めなさい。ただし、通信回線の伝送効率を50[%]とする。

2 次の文章を読んで、下の各間に答えなさい。(第32回)

初期のパソコン通信に使用されたモードムは300[baud]の規格のものが多かった。300[baud]は変調速度を表しており、当時の技術では変調速度とデータ信号速度は同じで、文字の通信さえも、伝送されてくる文字をディスプレイを使って目で追いかけられる程度の速度であった。やがて、モードムも高速化され、56000[bps]の規格のデータ転送速度のものが多く使用されるようになり、画像データなどの大容量のデータ伝送も実用的な速度となった。

- (1) 変調速度が300[baud]のモードムを使い、8ビットの文字コードを調歩式(スタートビット、ストップビット各1ビット)で送るとき、この文字コードのデータ転送速度[文字／秒]を求めなさい。ただし、モードムの変調速度とデータ信号速度は同じで、データ転送速度[文字／秒]は1秒間に何文字のデータを伝送できるかとする。また、回線のデータ信号速度はモードムのデータ信号速度と同じとし、回線の伝送効率を100[%]とする。
- (2) 350kバイトのデータを56000[bps]のデータ信号速度の回線で伝送したときに必要な時間は何秒か求めなさい。ただし、回線の伝送効率を51.2[%]とする。また、1kバイト=1024バイトとする。

3 次の文章を読んで、下の各間に答えなさい。(第33回)

データ通信において、変調速度とは、信号が1秒間に変調された回数をいい、単位には、[baud]が用いられる。位相変調(PSK : Phase Shift Keying)は、信号の位相を変化させてビット列を表現する方法である。4位相変調では、1回の変調の中で、たとえば、 45° ， 135° ， 225° ， 315° の4通りの位相を用いて、それぞれの位相に4通りの信号00, 01, 11, 10を対応させ、2ビット分の伝送をおこなっている。

- (1) 4位相変調において、変調速度が2400[baud]のとき、データ信号速度[bps]を求めなさい。ただし、[bps]とは、1秒間に伝送可能なビット数である。
- (2) 96kバイトのデータを4位相変調で2400[baud]の変調速度の回線で伝送したときに必要な時間は何秒か求めなさい。ただし、回線の伝送効率を80[%]とする。また、1kバイト=1000バイトとする。

4 9600bpsの通信回線を使って、8ビットの文字コードを調歩式で19200字送るための伝送時間[秒]を求めなさい。ただし、スタートビット、ストップビットは各1ビットとする。(第29回)

5 次の文中の空欄に最も適するものを解答群から選んで記号で答えなさい。(第28回)

伝送路を使って情報を伝送する場合に、受信装置と送信装置の動作のタイミングを合わせながら、文字などの情報を構成する各ビットを一定時間間隔で順々に送り出す方式を①という。送受信のタイミングを合わせる工夫の一つとして、文字を構成するビット群の前に②を、後ろに③を附加して伝送する方式がある。この方式を④という。また伝送線路において、文字またはその他のデータを構成するビット群を複数の線を使って同時に伝送する方式を⑤伝送、また1本の線に順次データをのせて伝送する方法を⑥伝送と呼んでいる。

解答群

- | | | | | |
|------------|-----------|---------|---------|------------|
| ア. 直列 | イ. 並列 | ウ. 接続 | エ. 開放 | オ. スタートビット |
| カ. ストップビット | キ. 調歩同期方式 | ク. 順次方式 | ケ. 同期方式 | |

6 512kバイトのデータを64k[bps]の容量の回線で伝送したときに必要な時間は何秒か求めなさい。ただし、回線の伝送効率を50[%]とする。

7 次の各文に最も関連の深いものを解答群から選びなさい。

- ①データ通信を行うときに、送信側と受信側であらかじめ決めておかなければならぬ規約。
- ②通信回線において、両方向伝送ではあるが、受信、送信を切り替えて片方ずつの伝送をおこなう通信方式。
- ③端末と電話回線などのアナログ回線の間に接続され、データ伝送の際、デジタル信号をアナログ回線に適した信号に変換し、また、その逆を行う装置。
- ④1組の直列に並んだデータ(ビット列)の前にスタートビット、後ろにストップビットを附加して送る同期方式。
- ⑤伝送すべきデータをある単位に分け、これに受信者、発信者名、誤り検出情報などをつけて送り出す通信方式。

解答群

- | | | | |
|------------|----------|----------|--------------|
| ア. 単方向通信方式 | イ. 半二重通信 | ウ. 全二重通信 | エ. プロトコル |
| オ. パケット通信 | カ. 調歩式 | キ. モデム | ク. ターミナルアダプタ |

8 64kbpsの通信速度の通信回線で、2秒ごとに平均2kバイトのファイルを転送するときの回線利用率(%)はいくらか。ただし、ファイルを転送する際に、元のファイル容量にくわえて、20%の容量の制御情報が付加されるものとする。

解答群

- | | | | | |
|--------|--------|--------|-------|-------|
| ア. 0.9 | イ. 6.3 | ウ. 7.5 | エ. 15 | オ. 30 |
|--------|--------|--------|-------|-------|

9 64kbpsの通信速度の通信回線で、768バイト/件の伝票データを5件ごとにまとめて、256バイトのヘッダ情報を附加して送りたい。1時間に平均して18,000件の伝票データを伝送するとき、回線利用率(%)はいくらか。ただし、1kビット=1024ビットとして計算する。

10 次の間に答えなさい。答は解答群の中から選びなさい。

- ①直列データ通信の同期式、非同期式の両方に適用する電気信号特性、機能特性、機械的特性などの規格で、パソコン通信などで多く用いられている規格を何というか。
- ②1組の直列に並んだデータ(ビット列)の前にスタートビット、後にストップビットを付加して送る同期方式を何というか。
- ③一つの通信回線で送・受信切り替えて通信を行う方式を何というか。
- ④コンピュータの出力を音声周波数に変調して送ったり、音声周波数に変調されたディジタル信号を復調してコンピュータが受けられるようにする装置を何というか。

解答群

ア. モデム	イ. 調歩同期式	ウ. RS-232C	エ. 半二重	オ. 全二重
--------	----------	------------	--------	--------

4. 2 インターネット向けのブロードバンドサービス

要 点

インターネットでデータサイズの大きな高品質の動画や音楽などをダウンロードする際に、ブロードバンドサービスを利用すると、広帯域通信回線を使用した高速なサービスを受けることができる。

①FTTH(Fiber To The Home)

光ファイバケーブルを一般家庭に引き込み、100Mbit/s程度の高速なデータ伝送を可能にする。メタリックケーブルを利用した通信回線と比較すると、基地局からの距離や雑音の影響を受けにくい。光ファイバケーブルを室内へ引き込み、回線終端装置をとおしてコンピュータに接続する。マンションなどの集合住宅で、個別の宅内への光ファイバケーブルの工事が困難な場合は、VDSLなどの技術と組み合わせて、共用部まで光ファイバケーブルで伝送し、各戸へは既存の電話線などを用いてVDSLで伝送するという方式もとられる。

②xDSL(x Digital Subscriber Line)

1. ADSL(Asymmetric DSL)

Asymmetricは非対称を意味し、上り(ユーザ→サーバ)と下り(サーバ→ユーザ)のデータ信号速度が異なる伝送方式を意味する。上りで0.5~1Mbit/s、下りで1.5~12Mbit/s程度（さらに高速なものもある）であり、一般的な電話回線やISDN基本インターフェースと比較して高速である。一般的な電話回線で使用している電話線（メタリックケーブル）を電話と共に使用することも特徴のひとつである。一般的な電話回線は4kHzまでの周波数帯域を使用するが、ADSLでは電話が使用していない4kHz以上の空きの周波数帯域を使用する。電話の音声通話の周波数帯域と全く異なるため、ひとつの回線に混在させることが可能となる。混在して送られる信号は、スプリッタという装置で音声信号とインターネット信号に分離し、更にADSLモデムを通してパソコンに送られる。

2. SDSL(Symmetric DSL)

Symmetricは対称を意味し、上りと下りのデータ信号速度が等しい伝送方式で、上り速度を重視する企業ユーザに利用されることを想定している。160kbit/s~2Mbit/s程度のデータ信号速度である。

3. HDSL(High-bit-rate DSL)

High-Bit-Rateは、高速伝送を意味し、1.5~2Mbit/s程度である。上りと下りのデータ信号速度が同じで、通信事業者などの幹線部分に使用されることを想定している。

4. VDSL(Very high-bit-rate DSL)

ADSLと同様に非対称な伝送方式である。下り(サーバ→ユーザ)が13~52Mbit/s程度で、上り(ユーザ→サーバ)が1.5~2Mbit/s。短距離で高速の伝送に適している。

③CATV(Cable TeleVision)

ケーブルテレビ(CATV)放送用に使用している光ケーブルや同軸ケーブルをインターネットのデータ通信回線として利用する。CATV回線にケーブルモデムを介してパソコンを接続する。理論的には、上り、下りとも32kbit/s~10Mbit/s程度、実際のサービスでは128kbit/s~1Mbit/sの通信が可能であるが、一定の帯域を近接のユーザと共有するためアクセスが集中すると通信速度が低下する可能性もある。100Mbit/s程度のものもある。

④ISDN(Integrated Services Digital Network)

サービス総合ディジタル網といい、従来、電話網、テレックス網、ディジタルデータ交換網などの通信網がそれぞれ独自に構築され、個々に加入者線を必要としてきたが、これら異なる回線を1本の加入者線で行えるようにした通信網である。基本インターフェースの構造は、一般に2B+Dと呼ばれ、2チャンネルのB(64kbit/s)と、1チャンネルのD(16kbit/s)から構成される。主としてBは、音声、データ等の情報の伝送に用いられ、Dは回線交換用の制御情報の伝送に用いられる。

□例題4.4□ 次の文章に最も適する語句を選択群から選びなさい。

- ①光ファイバケーブルを宅内へ引き込み、回線終端装置をとおしてコンピュータに接続し、高速なデータ通信を行う。
- ②上り(ユーザ→サーバ)と下り(サーバ→ユーザ)のデータ信号速度が非対称が異なる伝送方式で、既存の電話回線(メタリックケーブル)を利用した通信方式。
- ③サービス総合ディジタル網を用いたデータ通信を行う。
- ④マンションなどの集合住宅で、個別の宅内への光ファイバケーブルの工事が困難な場合は、既存の電話線などを組み合わせて伝送する場合もある。

解答群

ISDN	FTTH	ADSL	VDSL
------	------	------	------

解説

※要点参照

解答

- ①FTTH ②ADSL ③ISDN ④VDSL

練習問題

11 次の説明に最も適切な語を解答群から選びなさい。(第29回、第31回)

ADSL(AsymmetricDSL)において、Asymmetricは①を意味し、上り(②)と下り(③)のデータ信号速度が④伝送方式を意味する。一般的の電話回線で使用している電話線(メタリックケーブル)を用いて、音声信号とインターネット信号をひとつの回線に混在させることができ、混在して送られてくる信号は、⑤という装置で音声信号とインターネット信号に分離し、更に⑥を通してパソコンに送られる。

解答群

ア. 異なる	イ. 同じ	ウ. スプリッタ	エ. TA	オ. DSU
カ. プロバイダ	キ. 非対称	ク. 対称	ケ. ユーザ→サーバ	
コ. サーバ→ユーザ		サ. ADSLモデム		

12 次の説明に最も適切な語を解答群から選びなさい。

ディジタル通信網には、①多重の技術を利用して②交換方式と、③交換の技術を利用して④交換方式がある。②交換方式では、普通の電話機と同じように、データ伝送の必要が生じたときに端末間を交換機を通して電気的に②を設定する。回線の⑤に対して課金されるので、単位時間当たりのデータ量の多い通信に適している。④交換方式は、データを④と呼ばれる単位に分割して伝送する。各④にはヘッダ情報という送り先のコードが入っており、交換網は④ごとに送り先をチェックしながら③し、空いている回線を探して順次転送するので網を効率よく利用できる。データ量に対して課金されるため、⑤は長いがデータ量の少ない通信に適している。

解答群

ア. 蓄積	イ. 音声	ウ. 時分割	エ. 接続時間	オ. ADSL
カ. 回線	キ. ISDN	ク. パケット	ケ. パラレル	コ. シリアル
サ. プロトコル		シ. 総延長距離		

13 次の文中の [] 中に入るべき最も適切なものを解答群から選び記号で答えなさい。

ISDNは [①] と呼ばれ、その基本インターフェースの構造は、一般に2B+Dと呼ばれ、2チャンネルのB([②] kbit/s)と、1チャンネルのD([③] kbit/s)から構成される。主としてBは、音声、[④] 等の情報の伝送に用いられ、Dは回線交換用の制御情報の伝送に用いられる。

NT(ネットワーク終端装置)から交換局までの配線には、普通の電話網と同じ [⑤] メタリック平衡ケーブルを用いる。ただし、通常、この方式では半二重通信となるので [⑥] (2倍の速度で伝送して、一定時間ごとに伝送方向を切り替える双方向の全二重通信)を行う。

解答群

- | | | | |
|-----------------|-------------|----------------|---------------------------|
| ア. WAN | イ. データ | ウ. LAN | エ. サービス総合デジタル網 |
| オ. 8 | カ. 16 | キ. 32 | ケ. インターネットシステムディジタルネットワーク |
| コ. ISDNインターフェース | サ. 國際標準データ網 | シ. CAN | ス. エコー方式 |
| セ. 2線式 | ソ. 4線式 | タ. スーパーハイウェイ伝送 | チ. ピンポン伝送 |

5. ソフトウェアの基礎

5. 1 オペレーティングシステム

要 点

オペレーティングシステム(OS)は、コンピュータのシステムを管理し、コンピューターの利用者やアプリケーションソフトウェアが効率良くハードウェアを使用できるようにするためのソフトウェアで、基本ソフトウェアと呼ばれることもある。

1. オペレーティングシステムの目的

オペレーティングシステムの目的はシステム全体の生産性の向上であり、次のような目的がある。

- ① ハードウェア資源の有効利用
- ② 処理能力の向上（スループットの向上、ターンアラウンドタイムの短縮）
- ③ RASISの向上
- ④ 操作性・拡張性の向上

2. オペレーティングシステムの構成

オペレーティングシステムは、制御プログラムと処理プログラムで構成され、処理プログラムは、言語処理プログラムとサービスプログラム(ユーティリティプログラム)に分けられる。

このうちの制御プログラムの主な機能は、ジョブ管理、タスク管理、データ管理であり、この他に記憶管理・通信管理・運用管理・障害管理・入出力管理その他の機能がある。

(1) ジョブ管理

ジョブ管理の目的は、システムに投入したジョブを、自動的に連続して効率よく実行することであり、ジョブ管理の機能としては、スプーリング機能、スケジューリング機能、カタログ機能などがある。

ジョブは、マスタスケジューラとジョブスケジューラの2つの制御プログラムによって効率よく管理され、実行される。ジョブスケジューラは、リーダ、イニシエータ、ターミネータ、ライタという4つのプログラムによって構成される。

(2) タスク管理

タスク管理の目的は、同時に複数のジョブを並行処理することにより、CPUを効率的に使用することである。ジョブはジョブステップにわけられ、さらに処理の最小の単位であるタスクに変換され実行される。

タスクは生成されると、実行可能状態、実行状態、待ち状態の3つの状態を繰り返しながら処理され、実行が全て終わると消滅する。

(3) データ管理

データ管理は、ファイル管理とも呼ばれ、ファイル内のデータを総合的に管理する機能である。

(4) 記憶管理

記憶管理は、限られた記憶容量の主記憶領域にプログラムやデータを効率よく配置して利用できるようになる機能である。

記憶管理には、主記憶装置のみを扱う実記憶管理と、主記憶装置と補助記憶装置を組み合わせて仮想的な記憶装置として扱う仮想記憶管理がある。

(5) 通信管理

通信管理は、他のコンピュータや端末と通信回線を利用してデータ交換を行えるようにする機能である。

ネットワーク上のコンピュータが効率よく処理を行えるように制御する機能も持つ。

■例題5.1■ 次の文章の①～⑩に当てはまるもっとも適当な語句を解答群から選んで記号で答えなさい。

- (1) コンピュータに処理をさせる仕事の単位をジョブといい、ジョブの実行順序を効率よく管理するジョブ管理機能などは、オペレーティングシステムの中の①プログラムの基本的な機能である。
- (2) ジョブをさらに細かくそれぞれの作業単位に分けたものを②といい、一般にジョブは複数の②

から構成されている。

- (3) ジョブをコンピュータに与えるためには、③と呼ばれる言語が使用される。
- (4) ジョブ管理の機能には、④機能、スケジューリング機能、カタログ機能などがある。
- (5) ジョブ管理プログラムは、マスタスケジューラと⑤の二つの制御プログラムで構成されている。また、⑤は4つのプログラムで構成されており、そのうちのジョブの実行終了後の後始末を行うプログラムを⑥という。
- (6) 主記憶装置や⑦などのハードウェア資源をできるだけ効率よく利用するために、ジョブは⑧という最小の単位に変換して管理・実行される。
- (7) 複数の⑧を円滑に処理するためには、CPUの使用権を効率よく配分する必要がある。この働きをする制御プログラムを⑨と呼び、配分の方法としては、優先度方式、ラウンドロビン方式、タイムスライス方式、⑩方式などがある。

解答群

- ア. JCL イ. CPU ウ. BASIC エ. ロード オ. ジョブステップ カ. ジョブスケジューラ
キ. スポーリング ク. ターミネータ ケ. ディスパッチャ コ. ダイナミック サ. タスク
シ. 制御 ス. 処理

解説

①から⑧は、要点を参照。

タスクを効率よく処理するために、CPUの使用権を制御するプログラムをディスパッチャという。ディスパッチャによって複数のタスクを処理するシステムをマルチタスクプログラミングシステムという。CPUの使用権の与え方には、優先度方式、ラウンドロビン方式、タイムスライス方式、ダイナミック方式がある。

解答

- ①シ ②オ ③ア ④キ ⑤カ ⑥ク ⑦イ ⑧サ ⑨ケ ⑩コ

練習問題

1 次の文章に当てはまるもっとも適当な語句を解答群から選んで記号で答えなさい。(第29回)

- (1) 基本ソフトウェアは①プログラムと処理プログラムに大別される。
- (2) ジョブ管理機能のうち、プリンタなどの低速な入出力機器の入出力処理によって処理が遅れないように、メモリや磁気ディスクを仮想の入出力装置として利用して効率的なCPUの利用を行う機能を②機能という。
- (3) ジョブ管理機能のうち、システムに投入されたジョブを、その特性や優先順位に応じて実行順序を決めて効率よく処理する機能を③機能という。
- (4) ジョブスケジューラのプログラムのうち、入力待ち行列に登録されているジョブから、実行すべきジョブを取り出してジョブの実行準備を行い、ジョブの実行開始を指示するプログラムを④という。
- (5) ジョブスケジューラのプログラムのうち、ジョブの実行が終了した後の処理を行うプログラムを⑤という。

解答群

- ア. ディスパッチャ イ. イニシエータ ウ. カタログ エ. スケジューリング
オ. スポーリング ク. ターミネータ キ. 制御 ク. 監視

2 オペレーティングシステムに関する次の記述に当てはまる、もっとも適当な語句を解答群から選んで記号で答えなさい。(第28回)

オペレーティングシステムには次のような目的がある。

- 1.CPU、主記憶装置、補助記憶装置などの、ハードウェア①の有効利用を行う。
- 2.コンピュータシステムが単位時間あたりに処理する仕事の量を②といい、これを向上させる。
- 3.RASISのRは③を、Aは可用性、Sは保守容易性と④を、Iは⑤をさす。

この他に、コンピュータを使いややすくする操作支援などもオペレーティングシステムの目的のひとつである。

解答群

- | | | | | | |
|--------|--------|-----------|-------------|--------|--------|
| ア. ジョブ | イ. タスク | ウ. 汎用性 | エ. 機密性 | オ. 同時性 | カ. 信頼性 |
| キ. 保全性 | ク. 資源 | ケ. スループット | コ. ターンアラウンド | | |

3 次の記憶管理に関する記述の①～⑤に当てはまるもっとも適当な語句を解答群から選んで記号で答えなさい。

- (1) 制御プログラムの機能のうち、タスクに割り当てる記憶空間の管理を行うことを記憶管理という。記憶管理には、実際に存在する主記憶装置を効率的に利用するための①と、主記憶装置より大きなプログラムを実行できる②がある。
- (2) 実記憶装置の領域をあらかじめ区画に分けることはせず、未使用領域を見つけてプログラムをロードする方式を、可変区画方式または③方式と呼ぶ。
- (3) 主記憶装置と補助記憶装置の内容を入れ替えながら、優先順位の高いジョブを主記憶装置にロードする方式を④方式と呼ぶ。この方式は、ロールイン／ロールアウト方式とも呼ばれる。
- (4) プログラムをセグメントと呼ばれる単位に分割し、ルートセグメントを主記憶装置におき、補助記憶装置においてセグメントを必要に応じてロードする方式を⑤方式と呼ぶ

解答群

- | | | | |
|-----------|-----------|----------|----------|
| ア. 仮想記憶管理 | イ. 固定区画 | ウ. 動的再配置 | エ. 実記憶管理 |
| オ. オーバーレイ | カ. スワッピング | | |

5. 2 プログラム言語

要 点

1. プログラム言語の種類

プログラム言語の分類方法はいろいろあるが、まず大きく低水準言語と高水準言語に大別出来る。

(1) 低水準言語

機械語寄りの言語で機械語、アセンブラー言語などがある。

(2) 高水準言語

人が理解しやすい人間寄りの言語で、さらに手続き型言語と非手続き型言語に分類出来る。

手続き型言語は、問題解決の手順を記述するタイプのプログラム言語で、Fortran, COBOL, BASIC, Cなどがある。

非手続き型言語は、データを中心として処理を記述する言語で、処理の手続きそのものを記述する必要はない。手続き型言語に比べて開発の生産性もよい。代表的なものに、パラメータ記述言語のRPG, 関数型言語のLISP, 論理型言語のProlog, オブジェクト指向型言語のSmalltalk, C++, Javaなどがある。

2. 言語プロセッサ

プログラム言語で書かれた原始プログラム(ソースプログラム)を、機械語の目的プログラム(オブジェクトプログラム)に翻訳または変換したり、解読しながら実行させるプログラムを言語プロセッサという。

言語プロセッサには、アセンブラー、コンパイラ、インタプリタ、ジェネレータ、エミュレータ、シミュレータ、トランスレータなどがある。

□例題5.2□

次のプログラム言語に関する記述の①～⑨に当てはまるもっとも適当な語句を解答群から選んで記号で答えなさい。(第30回)

BASICやCOBOLや①などのプログラム言語は、コンピュータで行う処理の手順を中心として記述するので②言語と呼ばれる。これに対し、データを中心として処理の流れを組み立てていく言語を③言語と呼び、代表的なものに関数型のLispや論理型の④がある。また、最近ではデータとその操作を一体化

させた [5] を用意し、処理を [6] どうしのメッセージのやりとりとしてとらえる [7] という考え方方が広く使われるようになってきた。この考え方に基づいたプログラミングを [8] プログラミングといい、代表的なプログラム言語としては、[9] やC++などがある。

パソコンのエンドユーザが、手軽に処理を自動化するための簡易言語も発展してきている。表計算などで処理を自動化するために使用する簡易言語は [10] と呼ばれる。また、データベースの問い合わせには [11] が利用される。

解答群

- | | | | | |
|-------------|--------|-----------|--------------|-----------|
| ア. C | イ. SQL | ウ. Prolog | エ. Smalltalk | オ. オブジェクト |
| カ. オブジェクト指向 | キ. マクロ | ク. 手続き型 | ケ. 非手続き型 | |

解説

非手続き型言語の仲間であるオブジェクト指向型のプログラム言語が最近では広く用いられるようになってきた。オブジェクト指向は、現実の世界に存在するもの（オブジェクト）同士の関係をありのままに表現するために考え出されたもので、代表的な言語としてSmalltalkやC++, Javaなどがある。

表計算やワープロのソフトでは、エンドユーザ（利用者）がマクロを利用して簡単な処理を自動的に実行させることができる。また、SQLを使えば簡単にデータベースの問い合わせを行うことができる。これらの言語は、エンドユーザ言語と呼ばれる。

解答

- ①ア ②ク ③ケ ④ウ ⑤オ ⑥カ ⑦エ ⑧キ ⑨イ

解答

4 プログラム言語などに関する次の記述にそれぞれもっとも関連の深い語句を、解答群の中から選び、記号で答えなさい。（第22回）

- (1) コンピュータの内部では、利用者から依頼された仕事を能率よく処理するために、多くのプログラムがハードウェアやソフトウェアの制御・管理を行っている。これらのプログラムの集まりが [1] と呼ばれている。
- (2) 計算機システムの単位時間当たりの仕事の処理量を [2] といい、処理されたジョブの数やデータ量で表される。
- (3) 事務計算、経営科学計算、科学技術計算やデータベースシステムなど特定の処理を対象に作成されたプログラムを [3] という。
- (4) 「プログラムの構造定理」によると、どのような論理でも一つの入口と一つの出口を持つように設計すれば、順次・選択・ [4] の三つの制御構造で記述できる。
- (5) [5] は移植性に富む言語といわれており、EWS（エンジニアリングワークステーション）や小型機（ミニコン）を中心としたOS記述言語として使用されたことで有名である。
- (6) コンピュータの使用方式にはいくつかの種類があるが、その中で一つの処理装置において、二つ以上のプログラムを時間を区切って交互に実行させる使い方を [6] という。
- (7) 直接編成ファイルで各レコードにアクセスするとき、レコード中のキーの値をもとにして除算など一定の計算をしてアクセスすべきアドレスを指定する方法を [7] という。
- (8) プログラムを構成するモジュールの単体テストをボトムアップで行うとき、そのモジュールを呼び出す仮のモジュールを [8] と呼んでいる。
- (9) コンピュータのプログラムは知的生産物であり、財産的な利益は保護されなければならない。財産的な利益を保護する権利を [9] という。
- (10) 正常に動作しているソフトウェアを改変したり、悪質な場合はプログラムを破壊したり消去したりするような悪意のあるソフトウェアを、 [10] という。

解答群

- | | | | | |
|--------------|---------------|------------------|--------|--------|
| ア. マルウェア | イ. インターネット | ウ. オペレーティングシステム | | |
| エ. ハッキング | オ. 繰り返し | カ. ドライバ | キ. スタブ | ク. 著作権 |
| ケ. タイムシェアリング | コ. 終了 | サ. アプリケーションプログラム | | |
| シ. スループット | ス. マルチプロセッシング | セ. Fortran | ソ. C言語 | |

- 5 次のオブジェクト指向プログラミングについて述べた文の①～⑤に入るべきものを解答群から選び記号で答えなさい。(第34回)

従来からのFortranやCOBOLによるプログラミングは、処理の手順を記述するという①の手法である。オブジェクト指向によるプログラミングでは、処理手順ではなく処理対象そのものを中心と考える。オブジェクトとは、データとそれに関する手続きをひとまとめにしたもので、この手続きを②と呼ぶ。オブジェクト同士は③をやりとりしながら処理を進めていく。オブジェクトの内部の状態や手続きは外部から見えないようにになっており、これを④という。このようなオブジェクト指向プログラミングを行える言語としては、smalltalkや⑤などがある。

解答群

- | | | | | |
|----------|----------|---------|---------|-------|
| ア. クラス | イ. メッセージ | ウ. 手続き型 | エ. メソッド | オ. 繙承 |
| カ. カプセル化 | キ. システム化 | ク. C | ケ. C++ | |

- 6 次の記述にそれぞれ最も関係の深い字句を解答群の中から選び記号で答えなさい。(第26回)

- (1) BASICのように1行ごとに翻訳して実行するためのもの。
- (2) 表計算ソフト等で、一連の操作を記述し一括して実行した。
- (3) 32bitCPUのパソコンで8bitCPUのワンボードマイコンのオブジェクトモジュールを作った。
- (4) オブジェクトモジュールとライブラリモジュールを結合して実行モジュールを作成した。
- (5) 既存のプログラムを異なった種類のコンピュータで実行したい。そのために、元のシステムの環境を模倣するプログラムを使用し、その上でプログラムを実行した。

解答群

- | | | | | |
|---------------|--------------|-----------|----------|----------|
| ア. インタプリタ | イ. クロスアセンブラー | ウ. エミュレータ | エ. コンパイラ | |
| オ. コンパイラコンパイラ | カ. プリプロセッサ | キ. マクロ | ク. リンカ | ケ. コンバータ |

- 7 次の文章の①～⑩に当てはまる最も適当な語句を解答群から選び、記号で答えなさい。

- (1) Fortranやアセンブラー言語などのプログラム言語で書かれたプログラムを、コンピュータが実行できる①に翻訳するためのソフトウェアを②という。②には、アセンブラー言語で書かれたプログラムを翻訳する③や、FortranやCなどの高水準言語で書かれたプログラムを一括して翻訳する④や、BASICのように命令をひとつずつ翻訳して実行する⑤などがある。
- (2) プログラムのひな形が用意されており、目的の処理に合わせてデータやパラメータを入力すると自動的にプログラムを生成するソフトウェアのことを⑥といいう。
- (3) あるコンピュータの機能と同じ機能を、別の種類のコンピュータで実現するための機構を⑦といいう。従来は、ハードウェアとソフトウェアの組み合わせにより実現することが多かったが、最近ではCPUを含むコンピュータ能力の向上により、ソフトウェアだけで十分に実現できるようになってきた。
- (4) たとえば、8bitのCPU用のプログラムを開発するとき、より大きなコンピュータを使って開発した方が開発効率はよい。このように、あるシステム用のプログラムを別のシステムを使って開発することを⑧開発といい、そのためのコンパイラを⑧コンパイラといいう。
- (5) ソフトウェアを開発するとき、ソフトウェア開発の一連の作業を支援するソフトウェアを使うと、ソフトウェアの生産性を高めることができる。このような目的で使用されるソフトウェアを⑨ツールといいう。
- (6) コンパイラと逆の働きをする⑩を使うと、オブジェクト・プログラムから元のソースプログラムを生成することができる。このような技術を、リバース・エンジニアリングといいう。

解答群

- | | | | | | |
|--------------|------------|-----------|------------|---------|--------|
| ア. インタプリタ | イ. クロス | ウ. エミュレータ | エ. コンパイラ | オ. マクロ | |
| カ. ディスクコンパイラ | キ. プリプロセッサ | ク. ジェネレータ | ケ. ユーティリティ | | |
| コ. リンカ | サ. コンバータ | シ. アセンブラー | ス. 言語プロセッサ | セ. CASE | ソ. 機械語 |

5. 3 アプリケーションソフトウェア

要 点

1. アプリケーションソフトウェア

オペレーティングシステムなどの基本ソフトウェアに対し、業務処理を行うためのソフトウェアを応用ソフトウェアまたはアプリケーションソフトウェアという。

応用ソフトウェアには、特定の業務に使用するための個別応用ソフトウェアと、多くの業種や業務で共通に使える共通応用ソフトウェアがある。

2. ミドルウェア

基本ソフトウェアと応用ソフトウェアの中間のソフトウェアとして、支援ソフトウェアまたはミドルウェアと呼ばれるソフトウェアがある。

代表的なものとしては、グラフィックを利用することにより、ユーザがコンピュータとの対話処理をしやすいようにしたGUI（グラフィカルユーザインターフェース）、ソフトウェアの開発を支援するCASEツール、エンドユーザコンピューティングを支援するEUCツール、データベース管理システム(DBMS)、通信管理システムなどがある。GUIや通信管理システムは、最近ではOSの機能の一部となっているが、もとはミドルウェアである。

□例題5.3□ 次の記述の①～⑩に当てはまるもっとも適当な語句を解答群から選んで記号で答えなさい。

パソコンの普及により、OSやワープロソフトなどいろいろな種類のソフトウェアが市販されるようになった。これらのソフトウェアは、大型コンピュータ用の受託ソフトウェアに対して、ソフトウェア・パッケージと呼ばれることがある。一般に、ソフトウェア・パッケージはワープロなどの①をさすが、最近ではOSなどの②も頻繁に改良されるため、ソフトウェア・パッケージとして販売される。

ソフトウェア・パッケージを使用するときは、まずコンピュータに③する必要がある。また、ユーザの好みや使い方に合わせて、画面の設定やキーの機能などを変更したり設定し直したりすることもできる。この作業を④と呼ぶ。最近ではこれら一連の作業をセットアップと呼ぶこともある。

ソフトウェア・パッケージを使用するときは、一般に、ユーザはソフトウェアの使用条件を定める⑤に同意しなければならない。また、⑥を行うことにより、一定の期間アフターサービスを受けることができる。アフターサービスには、細かなバグを修正するための⑦や、電話やメールでの問い合わせのサポートなどがある。

また、インターネットで⑧することにより手に入れることのできるソフトウェアがある。これには、一定の条件が付いていたりするが、無料で使える⑨と、気に入って継続して使う場合には送金が必要な⑩がある。

解答群

- | | | |
|-------------|--------------|-------------|
| ア. 応用ソフトウェア | イ. フリーソフトウェア | ウ. 基本ソフトウェア |
| エ. シェアウェア | オ. インストール | カ. カスタマイズ |
| ク. ダウンロード | ケ. バージョンアップ | キ. 使用許諾契約 |

解説

パソコンの普及にともない多くのソフトウェアが市販されているが、これらはソフトウェア・パッケージと呼ばれる。基本的には、ワープロ、表計算などの応用ソフトウェアが中心であるが、パソコン用のOSなどの基本ソフトウェアも市販されている。この問題は、これらのソフトウェアに関する基本的な知識を問う問題である。

解答

- ①ア ②ウ ③オ ④カ ⑤キ ⑥コ ⑦ケ ⑧ク ⑨イ ⑩エ

8 次の説明にもっとも適当な語を解答群から選び記号で答えなさい。(第35回)

オペレーティングシステムなどの①に対し、業務を行うためのソフトウェアを②またはアプリケーションソフトウェアという。②には、特定の業務に使用するための③と多くの業種や業務で使える④がある。

①と②の中間のソフトウェアとして、⑤またはミドルウェアと呼ばれるソフトウェアがある。代表的なものとしては、ソフトウェアの開発を支援する⑥、通信システムやデータベース管理システム(DBMS)などがある。また、最近ではオペレーティングシステムの機能の一部となった⑦も、もとはミドルウェアである。

共通応用ソフトウェアは、ワードプロセッサソフトウェアや表計算ソフトウェアのように⑧として市販されるものが多い。これらのソフトウェアを使用するときには、まずコンピュータに⑨する必要がある。また、利用者の好みや使い方にあわせて、画面の表示やキーの機能などを設定することができ、これを⑩ともいう。

解答群

- | | | | |
|---------------|---------------|-------------|----------------|
| ア. アップデート | イ. カスタマイズ | ウ. インストール | エ. ソフトウェアパッケージ |
| オ. 共通応用ソフトウェア | カ. 個別応用ソフトウェア | キ. 応用ソフトウェア | ク. 支援ソフトウェア |
| ケ. 基本ソフトウェア | コ. 汎用ソフトウェア | サ. ダウンロード | シ. CASEツール |
| ス. GUI | ス. CUI | | |

9 次の文は何について説明したものか、その名称を解答群から選び記号で答えなさい。

- (1) 多くのデータを複数の利用者が効率よく、かつ有効に利用できるような形で集めたもの。
- (2) 人間とコンピュータの相互援助により、各種設計業務に使用する。建築設計、機械設計、LSIの設計などによく使われている。
- (3) 文章をディスプレイの上で作成したり、修正や編集ができる。また、その文章を印刷することができる。
- (4) ディスプレイ上に縦横に展開されるワークシートと呼ばれる集計用紙を用い、各種集計処理を行うことができる。
- (5) 遠隔地にある端末から、回線を通じて中央のコンピュータにデータを送り、まとめて処理をし、結果を回線を通じて端末に返送する。

解答群

- | | | | | |
|--------|------------|-------------|--------|----------|
| ア. 表計算 | イ. データベース | ウ. ワードプロセッサ | エ. TSS | オ. COMET |
| カ. CAD | キ. リモートバッチ | ク. LAN | | |

10 次の①～⑪に当てはまるもっとも適当な語句を解答群から選び記号で答えなさい。

A君たち5人のグループは、課題研究で環境破壊をテーマに選び研究することにした。近くに公立の中央図書館があり、コンピュータによって情報化されているので、そこで①システムを使って目的とする本を探すこととした。調べてみるとかなり多くの書物が発行されており、貸し出し中の本もかなりある。そこで次に、インターネットを使って情報を集めることにした。まず、検索データベースを使って目的とする情報を集めようとしたが、「環境破壊」という項目で検索するとかなり多くの件数が表示された。そこで、まず最初に環境破壊の中でも地球温暖化に注目し、ゴミ処理問題が原因と考えられるものから調べていくことにした。そこで、「環境破壊」だけでなく、「地球温暖化」、「ゴミ処理」などの語句全てが含まれる情報を検索することにした。このような方法を②といふ。

必要な情報を集めたら、情報を整理し分析する必要がある。このとき、書籍や新聞などの情報は、③を使って電子的な画像データに変換しておくと便利である。さらに、④ソフトを使えば、画像データを文字データに変換することも可能であり、後で情報を使いややすくなる。集めた情報のうち、数値的な情報については、⑤ソフトを使って集計や統計分析をし、必要に応じて⑥を作成しておくと情報がわかりやすくなる。また、データを⑦ソフトを使って⑦にしておけば、分類や検索が容易に行えるので便利である。また、インターネットで集めた情報の場合、わからないことがあれば、⑧を利用して質問すれば、有意義な回答を得られることがある。

少し前までは、情報を検討するときの報告書は、⑨ソフトを使って作成するのが一般的であった。

文字の大きさや **⑩** と呼ばれる字体を変えて読みやすくしたり、アンダーラインなどの文字飾りを使って強調することもできる。また、最近では文書の中に写真や図形だけでなく映像も張り付けることができるようになってきた。最近では、報告は **⑪** で発表を行うことが多くなってきた。この場合、**⑫** 支援ソフトで資料を作成すれば、配付資料も同時に作成できる。

解答群

- | | | | | |
|-------------|--------------|-----------|----------|----------------|
| ア. ワープロ | イ. 表計算 | ウ. データベース | エ. 文献検索 | オ. グラフ |
| カ. 絞り込み検索 | キ. バーコードリーダー | ク. OCR | ケ. OMR | コ. プрезентーション |
| サ. イメージスキャナ | シ. ダイレクトメール | ス. フォント | セ. 電子掲示板 | |

11 次の①～⑤に当てはまるもっとも適当な語句を解答群から選び記号で答えなさい。

表計算ソフトは、縦方向の **①** と、横方向の **②** からなる **③** によって、データの集計や計算を行うソフトウェアである。データはセルと呼ばれる単位で扱う。セルには数値や文字列などの値や、計算を行うための **④** も入力できる。計算や集計を簡単に行えるように多くの **⑤** が用意されている。

解答群

- | | | | | |
|--------|---------|-----------|---------|------|
| ア. 計算式 | イ. ブロック | ウ. データベース | エ. 行 | オ. 列 |
| カ. 関数 | キ. 方程式 | ク. ワークシート | ケ. クラスタ | |

12 次の①～⑤に当てはまるもっとも適当な語句を解答群から選び記号で答えなさい。

データベースはデータを効率よく利用するために考えられたものであり、データの間の関係をどのように捉えるかというデータモデルがいくつか考えられている。

代表的なデータモデルとしては、データの関係を木の形のようにとらえる **①** と、データの関係を網のようにとらえる **②** と、データを表の形で扱う **③** がある。

このうち、パソコンでよく使われるのは **④** を利用した **⑤** である。このデータベースでは、問い合わせ言語として **⑥** が用いられる。

解答群

- | | | | | |
|----------|--------------|---------|-------------------|----------|
| ア. 関係モデル | イ. ネットワークモデル | ウ. WEB | エ. SQL | オ. BASIC |
| カ. 階層モデル | キ. 階層データベース | ク. HTML | ケ. リレーションナルデータベース | |

13 次の①～⑤に当てはまるもっとも適当な語句を解答群から選び記号で答えなさい。

現在、パソコンの普及により、非常に多くのソフトウェアが市販されている。このような不特定多数の利用者を対象として、同じ目的で使用できるように商品化されたソフトウェアを **①** とよぶ。

会計処理や給与処理などの業務処理をシステム化するとき、これらのソフトウェアを利用すれば簡単にシステムを作ることが出来る。このように、業務や利用形態に合わせて、これらのソフトウェアの機能や仕様の一部を変更することを **②** という。このとき、開発を外注することも多く、注文に応じて **③** を利用した開発を専門に行う業者を **④** 業者という。これらの業者は、いくつかの **⑤** を組み合わせたり、**⑥** 独自の **⑦** でプログラムを作成したりしてシステムを構築する。これらのプログラムのうち、処理の手順を順番に記述するような方式は **⑧** と呼ばれることが多い。このように外注した場合でも、一から開発するのに比べて、比較的期間も短く、かなり安価にシステムを開発することができる。

解答群

- | | | | | |
|-----------------|---------|-----------|--------|-----------|
| ア. オペレーティングシステム | イ. WEB | ウ. VAR | エ. マクロ | オ. カスタマイズ |
| カ. ソフトウェアパッケージ | キ. 簡易言語 | ク. データベース | | |

5.4 ソフトウェアの開発

要 点

1. ソフトウェアの開発

ソフトウェアの開発工程として、いくつかのモデルが考えられている。

①ウォーターフォールモデル：もっとも基本的な開発方法で、工程をいくつかの段階に分けて、滝が上から下に流れるように、上位の工程から下位の工程へと順番に作業を進める方法である。

②プロトタイプモデル：開発の初期の段階でプロトタイプ（試作品）を作成することにより、要求仕様の確認や変更を行う方法である。

③スパイラルモデル：ウォーターフォールモデルとプロトタイプモデルの折衷モデルで、開発しようとするソフトウェアをいくつかの機能に分割し、その部分ごとに設計、プログラミング、テストを繰り返して完成していく方法である。

いずれのモデルでも、開発工程はきちんと管理し、それぞれの工程で文書化する作業も必要である。

また、テストも重要であり、様々なテストの方法がある。

□例題5.4□

次のウォーターフォールモデルによるソフトウェアの開発工程に順位を付けなさい。また、それぞれの工程に関係する作業内容を記号で選びなさい。

	工 程	順位	作業内容
①	基本計画	1	
②	プログラム設計		
③	テスト		
④	内部設計		
⑤	プログラミング		
⑥	外部設計		
⑦	運用・保守	7	

作業内容

- ア. プログラムをモジュールに分割する。
- イ. コーディングと単体テストを行う。
- ウ. 現状の問題点を分析し要求定義を行う。
- エ. ソフトウェアの動作確認を行う。
- オ. ソフトウェアの大まかな機能や構造を定義する。
- カ. ソフトウェアの詳細な機能や構造を定義する
- キ. ソフトウェアを使うためのマニュアルを整備する。

解説

ソフトウェア開発の基本的な開発技法はウォーターフォールモデルであるので、その手順と各工程の内容は重要である。工程の順序は、次の順である。

基本計画→外部設計→内部設計→プログラム設計→プログラミング→テスト→運用・保守

また各工程の内容は、解答を参照すること。

解答 順位 ①1 ②4 ③6 ④3 ⑤5 ⑥2 ⑦7
作業内容 ①ウ ②ア ③エ ④カ ⑤イ ⑥オ ⑦キ

14 ウォーターフォールモデルについての次の文章の①～⑩に当てはまる最も適当な語句を解答群から選び記号で答えなさい。

- (1) ウォーターフォールモデルによる開発工程において、基本計画から ① までの工程は、全体から部分へと作業を進めていくので、段階的詳細化または ② と呼ばれる。一方、テストの工程は部分から全体へと作業を進めるので、段階的統合化または ③ と呼ばれる。
- (2) プログラム設計において、ソフトウェアの機能をいくつかの機能に分割し、それらをさらに小さな機能に分割していく方法を ④ と呼ぶ。この方法は、プログラムを ⑤ と呼ばれる小さな単位に分割することにより、プログラムの見通しをよくできる。このとき、それぞれの ⑥ は、 ⑦ に基づいて分割され、 プログラミングされる。 ⑧ とは、「一つの入り口と一つの出口を持つプログラムはすべて、 ⑨ 、 ⑩ の 3 つの基本構造またはこれらの組み合わせによって表すことができる」というものである。

(3) プログラム言語を用いて、実際にプログラムを記述することを (10) と呼ぶ。

解答群

- | | | | | | | |
|----------------|----------|---------|----------------|-------|----------|----------|
| ア. 終了 | イ. 開始 | ウ. 順次 | エ. ジャンプ | オ. 選択 | カ. 構造化定理 | キ. 構造化設計 |
| ク. トップダウンアプローチ | ケ. モジュール | コ. 内部設計 | サ. ボトムアップアプローチ | | | |
| シ. コーディング | ス. 繰り返し | | | | | |

15 次の文章の①～⑩に当てはまる最も適当な語句を解答群から選び記号で答えなさい。

- (1) ソフトウェアを開発するとき、まずどのような(1)で、どのようなソフトウェアを作るのかといった(2)をたてる。次にこれにもとづいて実際に(3)を行う。出来上がったソフトウェアを(4)していく段階で、機能の改善や機能の追加が必要であればあれば改良が行われる。このときソフトウェアに(5)を付けておけば、改訂の履歴が分かり易い。細かい改良だけでは機能変更に対応出来なくなると、そのソフトウェアは(6)される。この一連の流れを、ソフトウェアの(7)という。
- (2) ソフトウェアの開発を行うとき日程計画は重要である。一般に日程計画は図表で表されることが多い。アローダイヤグラムと呼ばれる矢印を用いた図式で表す方法を(8)図という。また、作業者や作業を縦軸に、時間を横軸にとった日程表を(9)図という。
- (3) データがどのように流れて処理されるかを、データを中心として図式化したものを(10)という。

解答群

- | | | | | | | |
|---------|--------|--------|-------------|------------|----------|-------|
| ア. PERT | イ. DFD | ウ. TFT | エ. ガント | オ. バージョン | カ. 計画 | キ. 開発 |
| ク. 目的 | ケ. 運用 | コ. 廃棄 | サ. ターンアラウンド | シ. ライフサイクル | ス. システム化 | |

16 次の文章の①～⑩に当てはまる最も適当な語句を解答群から選び記号で答えなさい。

ウォータフォールモデルによるソフトウェア開発工程において、テストはまず(1)から行われる。これは、作成したモジュールがうまく動作するかどうかのテストである。

次に、モジュールどうしを結合させて(2)を行う。このとき、上位モジュールから下位モジュールへと順番に結合していくテスト方法を(3)という。このとき、未完成の下位モジュールの代わりをするプログラムを(4)と呼ぶ。

また、下位モジュールから上位モジュールへと言う順序で行うテスト方法を(5)と呼ぶ。このとき、未完成の上位モジュールの代わりをするプログラムを(6)と呼ぶ。この他にすべてのモジュールをすべて結合して行う(7)という方法もある。

一応完成したソフトウェアについて、機能・性能・操作性などが、要求仕様を満たしているかどうかと言うテストを総合的にテストする。このテストを、総合テストまたは(8)という。

最終テストは運用テストと呼ばれ、実際に運用する時と同じ環境やデータでテストを行う。

これらのテストのうち、プログラムの内部構造や論理に基づいて行うテストを(9)と呼び、(10)がこの方法で行われる。一方、プログラムの内部構造は分からぬものとして、プログラムの仕様だけからテストする方法を(11)という。(12)以降のテストはこの方法で行われる。

解答群

- | | | | |
|----------------|----------|----------------|-----------|
| ア. ボトムアップテスト | イ. 折衷テスト | ウ. ビッグバンテスト | エ. 結合テスト |
| オ. テストデータ | カ. スタブ | キ. システムテスト | ク. ドライバ |
| コ. ブラックボックステスト | サ. 単体テスト | シ. ホワイトボックステスト | ス. オブジェクト |

6. その他の情報関連知識

6. 1 情報技術と社会

要 点

現代社会は、高度情報通信社会とよばれる。携帯電話やパソコンなどの情報機器やインターネットをはじめとする情報通信システムにより、必要とする情報を必要なときに素早く手に入れることが出来るようになってきた。この分野は、解説する範囲が広いので、個々の語句について説明しないが、例題や問題を通じて学習してほしい。

□例題6.1□ 次の語句と最も関係のある英略字を解答群から選び記号で答えなさい。

- (1) 地域気象観測システム
- (2) コンピュータ支援設計
- (3) 販売時点情報管理システム
- (4) コンピュータ支援学習
- (5) 生産自動化
- (6) 戦略情報システム
- (7) 自動受発注システム
- (8) コンピュータ統合生産
- (9) 高度道路交通システム
- (10) 電子商取引

解答群

ア. AMeDAS	イ. CAD	ウ. CAI	エ. CAM	オ. CATV	カ. CIM	キ. DTP
ク. EC	ケ. EOS	コ. FA	サ. ITS	シ. POS	ス. SIS	

解説

コンピュータの世界では略語がよく使われる。技術革新の激しい分野なので、同じ略語が違う意味で使われるようになることがある。また、分野が違えば同じ略語が違う意味で使われることもある。たとえば、「PC」は情報の世界では「Personal Computer」であるが、制御の世界では「Programmable Controller」であったりする。ここでは、比較的よく使われる略語を取り上げた。

- (1) Automated Meteorological Data Acquisition System
- (2) Computer Aided Design
- (3) Point Of Sales System
- (4) Computer Assisted Instruction
- (5) Factory Automation
- (6) Strategic Information System
- (7) Electronic Ordering System
- (8) Computer Integrated Manufacturing
- (9) Intelligent Transport Systems
- (10) Electronic Commerce

解答 (1) ア (2) イ (3) シ (4) ウ (5) コ (6) ス (7) ケ (8) カ (9) サ (10) ク

練習問題

1 次の記述中の①～⑩に入れるべき適当なものを解答群の中から選びなさい。

- (1) ①は、自動受発注システムの略称で、発注すべき商品のコードと②をデータ入力端末で入力し、③で仕入れ先に送信し、自動発注したり受注したりするシステムである。このシステムを導入すると、取引先とのデータの授受が自動化されるので、迅速・正確な受発注ができるようになる。

- (2) 銀行における現金自動支払い機（④と略される）や現金自動預け払い機（⑤と略される）は、金融機関ネットワークにおいて入出金業務の効率化と⑥の向上を目的として実用化された。顧客と金融機関とのネットワークには、ファームバンキングとホームバンキングがある。
- (3) ⑦システムは、販売時点情報管理システムの略称で、販売の時点で商品に印刷されている⑧を、レジに付属している⑨で読み取り、商品名、価格などの情報を即時に入力して処理する方式である。この時同時に⑩なども入力し、販売管理に利用されている。

解答群

ア. オンライン	イ. 自動化	ウ. 数量	エ. 顧客情報	オ. 顧客サービス
カ. バーコード	キ. スキャナ	ク. POS	ケ. EOS	コ. ATM サ. CD

- 2 次の①～⑤に当てはまる適当な語句を解答群の中から選び、記号で答えなさい。

コンピュータなどの情報機器を積極的に利用する活動を総称して情報化という。社会生活でも、コンピュータが盛んに利用されるようになってきた。たとえば、天気予報は①と呼ばれる、気象資料総合処理システムの情報をを利用して行われている。その基となるデータは、全国1300以上の地点の無人の自動気象観測所から通信でセンターに送られる。この地域気象観測システムは、②と呼ばれる。また、雲の様子など地球をとりまく環境については③システムにより情報が送られてくる。

また、④と略される汎地球測位システムは、4個以上の専用の人工衛星からの電波を同時に受けて、船舶・航空機・自動車などが自分の⑤を知るシステムである。現在では、2地点間の相対位置を水平距離で1cm、高度差で数cmの精度で測定でき、身近なところではカーナビゲーションシステムなどに使われている。

解答群

ア. ADSL	イ. GPS	ウ. AMeDAS	エ. COSMETS	オ. ISDN
カ. 大きさ	キ. プラント	ク. ひまわり	ケ. 速度	コ. 位置

- 3 次の①～⑤に当てはまる適当な語句を解答欄から選び、記号で答えなさい。

産業界ではコンピュータを積極的に利用することにより情報化を進展させた。事務処理分野では計算処理の自動化やワープロの導入などが行われたがこれらを①と呼ぶ。また、生産現場では製造工場の無人化や自動化を図る②と呼ばれる情報化が行われた。これは、コンピュータを使って設計製図を行う③や、そのデータを使って自動的に製造を行う④などのシステムが使われている。

最近、利用者の製造に対する要求は多様化しており、従来の大量生産方式では対応できなくなっている。したがって、多品種・少量生産を自動的・効果的に行う生産システムが開発された。これを⑤という。

解答群

ア. AO	イ. TM	ウ. CAI	エ. CAD	オ. CAS
カ. CAM	キ. DC	ク. FA	ケ. FMS	コ. OA

6. 2 ネットワーク

要 点

1. コンピュータネットワークの種類

- (1) LAN (Local Area Network) : 同一建物内や同一敷地内などの限られた範囲のコンピュータをそれぞれ接続し、資源や情報の共用を目的としたコンピュータネットワークシステム。
- (2) WAN (Wide Area Network) : 広域ネットワークで、LANを専用線や公衆回線で接続したもの。
- (3) インターネット(internet) : 世界規模のWANで、コンピュータ・ネットワークの集合体。最近では、コンピュータはインターネット接続できることが基本となっており、郵便や電話網と同じくらい重要な情報ネットワークである。

(4) イントラネット(Intranet)：「Intra」は「内部の」という意味の接頭語で、インターネットの技術やインフラを使って構築した企業などの組織内のコンピュータネットワークをいう。インターネットの技術を利用することにより、ハードウェアとソフトウェアの両面で経費の削減を行うことが出来る。また、関係企業などのイントラネットを相互に接続したネットワークをエクストラネット(Extranet)、「Extra」は「外部の」という意味の接頭語)という。

2.LANの種類

- (1) クライアント・サーバシステム：クライアントがサーバに要求を出しながら作業を進めていくタイプのネットワーク形態。サーバには、ファイルを共有管理するためのファイルサーバ、プリンタ(印刷)を共有管理するためのプリンタサーバ、データベースのデータを供給管理するためのデータベースサーバなど多くのサーバがある。
- (2) ピアツーピアシステム：クライアントサーバシステムと異なり、コンピュータ間に固定された主従関係のないネットワーク形態。最近のOSは標準でネットワークを構築する機能を持っており、複数のコンピュータをピアツーピアシステムとして接続し、簡単にファイルやプリンタの共用をすることができる。

3.プロトコル

コンピュータによるデータ通信を行うために必要な通信のための取り決めを通信プロトコルという。コンピュータ同士を比較的自由に接続出来るようになったのは、OSI参照モデルという標準ネットワークプロトコルが提唱されたからである。現在では、インターネットの普及にともない、インターネットの標準プロトコルであるTCP/IP(Transmission Control Protocol/Internet Protocol)が、パソコンなどのコンピュータの標準プロトコルとなっている。TCP/IPのTCPはOSI基本参照モデルの第4層、IPはOSI基本参照モデルの第3層に相当するプロトコルである。

OSI参照モデル		TCP/IPのプロトコル
第7層	アプリケーション層	アプリケーション プロトコル (ファイル転送) (仮想端末など)
第6層	プレゼンテーション層	TCP
第5層	セッション層	IP
第4層	トランスポート層	サブネットの プロトコル (イーサネットなど)
第3層	ネットワーク層	
第2層	データリンク層	
第1層	物理層	

4.インターネットのサービス

- (1) WWW (World Wide Web) : Webページを閲覧する機能。HTML(Hypertext Markup Language)で書かれたハイパーテキストにより、リンクをたどって情報を得ることが出来る。Webページを閲覧するためには、ブラウザと呼ばれるソフトウェアが必要である。
- (2) 電子メール : 文字通り、電子的なメールをやりとりする機能。送信にはSMTP(Simple Mail Transfer Protocol)というプロトコルが使われ、受信にはPOP3(Post Office Protocol Version 3)というプロトコルが使われる。最近ではPOP3より柔軟性と拡張性に優れたIMAP(Internet Messaging Access Protocol)というプロトコルが使われ始めている。メールの送受信には、メールと呼ばれるソフトウェアが必要である。
- (3) FTP (File Transfer Protocol) : ファイル転送プロトコル。これにより、インターネット上のファイルをダウンロードしたり、自分の作ったホームページをアップロードしたり出来る。
- (4) ネットニュース(Net News) : インターネット上の電子掲示板サービス。インターネット上に点在するNNTP (Network News Transport Protocol)サーバが相互接続して、データを転送するように分散管理している。ニュースを読むためには、ニュースリーダーと呼ばれるソフトウェアが必要である。
- (5) Telnet : TCP/IPネットワークにおいて、仮想端末を実現するためのプロトコルおよびプログラム。遠隔地のコンピュータを自分のキーボードなどで操作出来る。

■例題6.2■

次のネットワークに関する次の記述中の①～⑩に入れるべき適当なものを解答群の中から選びなさい。(第36回)

OSIは、①システム間相互接続と呼ばれ、②が制定したネットワークアーキテクチャである。

OSI基本参照モデルとして、機能・体系別に7層に分けられたモデルが提案されており、データ通信システムの基本となっている。

OSI基本参照モデルは、最下層から順に、物理層・③層・④層・⑤層・⑥層・プレゼンテーション層・⑦層と呼ばれる。

インターネットの標準プロトコルであるTCP/IPは、OSI基本参照モデルを基礎としている。OSI基本参照モデルとTCP/IPを対応させると、OSI基本参照モデルの物理層を第1層とすると、TCPはOSI基本参照モデルの第⑧層に対応する層に属し、IPは第⑨層に対応する層に属する。また、TCP/IPでは、データは⑩と呼ばれる小さな単位に分割されて送られる。

解答群

- | | | | |
|-------------|-----------|------------|----------|
| ア. アプリケーション | イ. ネットワーク | ウ. インターネット | エ. セッション |
| オ. データリンク | カ. パケット | キ. トランスポート | ク. 循環型 |
| コ. ISO | サ. JIS | シ. ANSI | ス. 2 |
| | | | セ. 3 |
| | | | ソ. 4 |
| | | | タ. 5 |

解説

OSI基本参照モデルとTCP/IPはネットワークの基本であるので、要点を参考にして必ず覚えておいてほしい。特に、4層で構成されるTCP/IPは、インターネットの標準プロトコルであり、パソコンを中心とした小規模なLANをはじめ、インターネットとの接続を前提にしたネットワークの標準プロトコルになりつつある。

TCP/IPでは、データはパケットと呼ばれる小さな単位に分割されて、送信先や送信元などのデータの入ったヘッダを付けて送られる。

解答

- ①ケ ②コ ③オ ④イ ⑤キ ⑥エ ⑦ア ⑧ソ ⑨セ ⑩カ

練習問題

4 次の文をよく読んで、解答群にある語句の中から対応するものを選び、記号で答えなさい。

- (1) インターネットにおいて、ファイル転送には、①というプロトコルが用いられる。
- (2) インターネットにおいて、電子メールの送信には②というプロトコルが、受信には③というプロトコルが用いられる。
- (3) インターネットのWebページは④と呼ばれる言語で書かれており、表示するためには、⑤と呼ばれる閲覧ソフトウェアが必要である。

解答群

- | | | | | | |
|------------|-----------|---------|---------|---------|---------|
| ア. CSMA/CD | イ. HTTP | ウ. FTP | エ. POP3 | オ. FTTH | カ. SMTP |
| キ. HTML | ク. Telnet | ケ. パケット | コ. ブラウザ | サ. メーラー | |

5 インターネットに関する次の文章の①～⑩に入れるべき適当なものを解答群の中から選び、記号で答えなさい。

- (1) WWWサーバに置くWebページと呼ぶ情報は、①と呼ばれる言語で書かれている。利用者は、②と呼ばれる閲覧ソフトを使って、文字、音声、静止画、動画などの③からなる情報を見ることができる。
- (2) インターネット上のホームページなどのオブジェクトの位置は、④で指定する。例えば、全国工業高等学校長協会のホームページの④は、「http://www.zenkoukyo.or.jp/」である。「http:」の部分は、アクセスのためのプロトコルを表しており、この場合、⑤を転送するためのプロトコルを表している。ファイル転送の場合は、「http」の部分が⑥となる。
- (3) インターネットに接続されたコンピュータには、世界中で重複しない⑦が付けられている。これは現在は、「.」で区切られた、0～⑧までの10進数の4つの数字の組み合わせで表される。これを人間に理解しやすいように表したのが、⑨であり、⑩サーバにより対応付けがなされている。

解答群

- | | | | | | |
|-----------|------------|------------|----------|-------------|---------|
| ア. ファイル | イ. ブラウザ | ウ. マルチメディア | エ. ドメイン名 | オ. ハイパーテキスト | |
| カ. IPアドレス | キ. TCPアドレス | ク. URL | ケ. HTML | コ. DNS | サ. SMTP |
| シ. nntp | ス. ftp | セ. 128 | ソ. 255 | | |

6 次のLANのアクセス方式のうち、複数の端末が同時に送信を行い、送信の衝突が起きる可能性のあるものとないものとに分けなさい。

- (1) 衝突が起きる可能性のあるもの ()
(2) 衝突が起きる可能性のないもの ()

解答群

- | | |
|---------|------------|
| ア. ATM | イ. CSMA/CD |
| ウ. FDDI | エ. トーカンリング |

7 ネットワークに関する次の記述で①～⑧に入る適切な語句を解答群から選び、記号で答えなさい。

社内のネットワークシステムを、インターネットの技術やインフラを使って構築したものを①と呼ぶ。WWWブラウザなどインターネット用のソフトウェアが使えるので、独自のソフトウェアを用意しなくてよいので経済的である。

社内のネットワークシステムを、インターネットに接続する場合、外部からの不正なアクセスを防止するために②を設置する。一般に②の内側でしかも内部のネットワークからも切り離された部分に、外部公開用のWWWサーバやメールサーバを置く。この範囲は③とよばれ、インターネットからの不正なアクセスを防ぐとともに、内部ネットワークへの被害の拡散も防止する。しかし、いくらこれらの仕組みがあっても、外部から④の入ったファイルをフロッピーディスクなどで持ち込めば、システムが破壊されることもある。また、利用者が気付かないうちに情報を外部に送信する⑤と呼ばれるソフトウェアがインストールされることもある。さらに、ユーザーごとに⑥を制限しても、他人のユーザIDと⑦を使ってその人になりますました場合、情報を盗まれたり破壊されたりすることがある。これらになりすましを防止するために、指紋などの⑧が使われ始めている。

解答群

- | | | | | |
|-------------|---------|---------------|-----------|----------|
| ア. パスワード | イ. ワクチン | ウ. コンピュータウイルス | エ. DMZ | オ. アクセス権 |
| カ. ゲートウェイ | キ. 生体認証 | ク. ファイアウォール | ケ. ルータ | コ. WAN |
| サ. エクストラネット | | シ. イントラネット | ス. スパイウェア | |

8 ネットワークに関する次の記述にあてはまる、最も適切なものを解答群から選び、記号で答えなさい。

- ①ネットワークの通信を可能にするために、コンピュータの拡張スロットに取り付ける機器を何というか。
②通信距離を伸ばすために、ケーブルの途中に入れて信号を增幅する装置で、OSI基本参照モデルの第1層に対応する。
③ツイストペアケーブルを数本接続するための集線装置で、小規模なLANならこれを複数接続してピアツーピアネットワークを構築出来る。
④異なる種類のLANやLANとWANの接続に用いられ、複数のLANを接続したときに、最適な経路を選択してデータの通る経路を設定する機能を持つ。OSI基本参照モデルの第3層以下に対応する。
⑤通信プロトコルなどの接続方式の異なるネットワークを接続する装置。LANとWANやWANとWANの接続に用いられ、伝送元のプロトコルから伝送先のプロトコルへと、データの変換を行うソフトウェアも含めた装置を指す。OSI基本参照モデルの7層全てに対応する。
⑥同じ種類のLANの接続に用いられ、種類の異なったケーブルの接続などを行える。OSI基本参照モデルの第2層以下に対応する。

解答群

- | | | | | | |
|-----------|---------|----------|---------|--------|--------|
| ア. ゲートウェイ | イ. ブラウザ | ウ. ルータ | エ. ケーブル | オ. NIC | カ. HUB |
| キ. ブリッジ | ク. リピータ | ケ. ターミナル | | | |

6. 3 情報処理システム

要 点

情報処理システムを分類するとき、データの処理方式によって分類する方法と、システムのハードウェア構成によって分類する方法がある。これらは相互に関係しあっているので、明確に分けられるものではない。代表的なシステムについて列挙する。

1. データの処理方式

コンピュータシステムはデータの処理方式で、バッチ処理とリアルタイム処理に大きく分類出来る。

(1) バッチ処理：処理するデータやプログラムを一時蓄積しておき、適当な時間に一定量の処理をする方式で一括処理方式ともいう。

a) ローカルバッチ処理：通信回線を使用しないバッチ処理を、ローカルバッチ処理という。

b) リモートバッチ処理：通信回線を使って、オンラインでデータをセンターに送ってバッチ処理を行う場合を、リモートバッチ処理という。

(2) リアルタイム処理：端末でデータ処理が必要になったらその都度コンピュータで処理され、結果も直ちに得られるシステムで、実時間処理、即時処理とも呼ばれる。オンラインで行えば、オンラインリアルタイム処理と呼ばれる。

a) 対話型処理：バッチ処理に対しての呼ばれ方で、パソコンでは当たり前的方式。キーボードなどの入力装置からデータや命令を入力すると、ディスプレイ装置などから出力が返ってくる処理方式。汎用機で会話処理を実現するために、コンピュータを非常に短い時間で区切って時分割で使用し、各端末は1台のコンピュータを占有しているように使えるシステムをタイムシェアリングシステムという。

2. 集中処理システムと分散処理システム

1台のコンピュータで集中して処理を行うか、複数のコンピュータで処理を分散して行うかによって、集中処理と分散処理に分類出来る。

(1) 集中処理システム：複数の端末を中央の大型コンピュータに通信回線などで接続し、処理を全て中央のコンピュータで行わせるシステム。ハードウェア構成で次のような種類がある。

a) シンプレックスシステム

b) デュプレックスシステム

c) デュアルシステム

d) マルチプロセッサシステム

(2) 分散処理システム：複数のコンピュータをネットワークで接続し、それぞれのコンピュータが処理を分担して、全体として効率よく処理を行っていくシステム。次の2つの種類がある。

a) 垂直分散処理システム

b) 水平分散処理システム

例題6.3

あるコンピュータシステムで、平均故障間隔が475時間、平均修理時間が25時間であった。このとき、このシステムの稼働率を求めなさい。

解説

システムの稼働率は、次式で求めることができる。

$$\text{稼働率} = \frac{\text{平均故障間隔}}{\text{平均故障間隔} + \text{平均修理時間}}$$

解答

$$\text{稼働率} = \frac{475}{475+25} = 0.95$$

練習問題

9 コンピュータシステムの処理方式に関する次の記述中の、①～⑤に入れるべき最も適当なものを解答群の中から選び、記号で答えなさい。

- (1) コンピュータシステムの分類の一つとして処理方式による分類があるが、① 処理方式は、ある期間に集められたデータやプログラムをまとめて処理する方式であり、② 処理方式は、処理要求やデータを発生のつど入力し、その処理結果を、そのときのシステムの環境の働きに充分間に合うだけ速やかに出力する方式である。
- (2) ③ は、コンピュータが適当な時間刻みで複数の処理を交互に行い、ユーザはあたかもコンピュータを独占しているかのように利用できる方式である。また、④ 処理方式は、それぞれの使用場所に独自のコンピュータを置き、これで大半のデータ処理を行うが、他の場所と関連するデータはセンタのコンピュータで処理する方式である。
- (3) ⑤ 処理は、オンラインシステムの端末を使用して行う処理で、端末側に蓄積されたデータやプログラムをセンターに送って一括処理するので応答時間が短縮される。

解答群

- | | | | |
|------------------|------------|-------|--------|
| ア. タイムシェアリングシステム | イ. リモートバッチ | ウ. 分散 | エ. バッチ |
| オ. リアルタイム | | | |

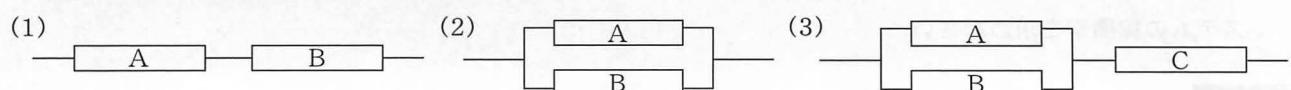
10 RASISに関する次の記述に当てはまる、もっとも適当な語句を解答群から選んで記号で答えなさい。

- (1) Rはreliabilityの頭文字で、① と呼ばれる。これは、システムが安定して動作する目安で、指標として故障から故障までの間隔の平均である② が用いられる。
- (2) Aはavailabilityの頭文字で、③ と呼ばれる。これは、コンピュータが正常に動作している割合を示し、指標として④ が用いられる。
- (3) Sはserviceabilityの頭文字で、⑤ と呼ばれる。システムに障害が発生したときの修理のしやすさを示し、指標として修理に要する時間の平均である⑥ が用いられる。
- (4) Iはintegrityの頭文字で、⑦ と呼ばれる。これは、ハードウェアの故障およびソフトウェアの異常や故意によるデータの破壊を防止できること。また、もしもデータが破壊されても容易に修復できること。
- (5) Sはsecurityの頭文字で、⑧ と呼ばれる。これは、情報の漏洩や破壊などを防ぎ、データの保護や機密保持を行うこと。
- (6) システムに障害が発生したときに、正常な動作を保ち続ける能力をフォールトトレランスという。これは、システムの一部が故障しても、全体対としての能力を下げても引き続き動作を可能にする⑨ と、安全を重視し、故障による被害を最小限に抑えることを重視した動作を行う⑩ がある。

解答群

- | | | | | | |
|----------|--------|------------|------------|---------|---------|
| ア. 保守容易性 | イ. 汎用性 | ウ. 機密性 | エ. 同時性 | オ. 信頼性 | カ. 保全性 |
| キ. 可用性 | ク. 稼働率 | ケ. フェイルセーフ | コ. フェイルソフト | サ. MTTR | シ. MTBF |

11 コンピュータが次の図の様に接続されたシステム全体の稼働率をそれぞれ求めなさい。ただし、全てのコンピュータの稼働率は0.9とする。*1



12 常時稼働させることの必要なあるシステムがある。このシステムの平均故障間隔は1980時間で、故障が発生したときには平均して修理に20時間かかっていた。そこで、995時間に1回の割合で保守点検をすると、故障が起らなくなった。ただしこの保守点検にはシステムを平均5時間停止しなければならない。このシステムについて、保守点検を行わない場合と、保守点検を行う場合のそれぞれの稼働率を求めなさい。

*1(問題5.11)コンピュータを直列に接続したときのシステム全体の稼働率 $O = A \times B$

コンピュータを並列に接続したときのシステム全体の稼働率 $O = 1 - (1 - A) \times (1 - B)$

13 次の記述はオンラインシステムについて述べたものである。最も適当なシステムの名称を解答群の中から選び、記号で答えなさい。

- (1) C P Uを1台だけ使用し、比較的小規模なシステムに用いられ、経済的だが一度障害が発生するとシステム全体がダウンしてしまうシステム。
- (2) 主系と従系の2つの処理系を持ち、通常は主系でオンライン処理を行い、従系ではバッチ処理を行うシステム。主系に障害が起きたときは、従系をオンライン処理に変更し、システム処理を維持する。
- (3) 並列システムとも呼ばれ、2組のコンピュータシステムで同一の処理を行い、常に処理結果を照合しながら処理を進めるシステムで、一方に障害が発生しても処理能力を低下させることなく継続できる。高い信頼性が必要な列車制御システムや航空管制システムなどに利用される。
- (4) 処理装置を複数台接続し、それぞれに処理機能を分担させてシステム全体の処理能力を向上させたシステム。ただし、通信装置、記憶装置は共用するので、チャネルという入出力制御装置が必要である。

解答群

- | | | |
|----------------|----------------|-----------------|
| ア. タンデムシステム | イ. デュプレックスシステム | ウ. デュアルシステム |
| エ. シンプレックスシステム | オ. マルチメディアシステム | カ. マルチプロセッサシステム |

14 システムに要求される特性およびそれを満足するシステム形態に関する次の記述のそれぞれについて関連の深いものを解答群から選びなさい。(第28回)

- ①高い信頼性とリアルタイム性を要求されるシステムであり、2組のシステムが完全に並列運転を行い、それぞれが相互監視する。一方のシステムに障害が発生したときは、これを全体の系から切り離して運転を続行する。
- ②高い信頼性を要求されるシステムであり、オンライン処理を行う現用系（主系）と予備系から構成されるが、①のシステムほどリアルタイム性は必要とされず、通常は予備系で他のジョブを実行してもよい。
- ③事務計算、技術計算いずれにおいてもCPU使用率の高いジョブが多いため、入出力に比べCPU処理がネックとなり、システムの効率が上がらないとき使用するシステム。
- ④ユーザ部門が同一事業所内に分散しており、それぞれの部門に必要な小規模システムを配備してデータ処理を行いたいときに使用するシステム形態。ただし、データは部門間で共有することもあり、ファイルのオンライン高速転送は不要である。
- ⑤中央の計算機に遠隔地の小型計算機または端末からデータまたはプログラムを含むデータ処理入力情報を全体を伝送し、中央の計算機側で処理を行うシステム。

解答群

- | | | |
|----------------|----------------|------------------|
| ア. シンプレックスシステム | イ. デュアルシステム | ウ. マルチプロセッサシステム |
| エ. デュプレックスシステム | オ. リモートバッチシステム | カ. ローカルエリアネットワーク |

6.4 マルチメディア

要 点

マルチメディアの厳密な定義は決まっていないが、一般に、文字、音声、静止画、動画などの様々な情報伝達手段となりうるメディアを複合的に扱うこと、あるいは扱う技術や機器、ソフトウェアなどをいう。従来の、新聞、ラジオなどは単一のメディアであったが、現在のインターネットはマルチメディアの典型である。

1. メディアの種類

- (1) 文字：ワープロを使うことにより、文字のフォントや色、大きさなどを自由に変えることにより、より豊かな表現が出来るようになった。
- (2) 図表：表計算ソフトには多種なグラフを作成する機能が付いており、説得力のあるレポートを作成出来る。
- (3) 画像：画像作成ソフトウェアにはペイント系とドロー系があり、目的により使い分ける必要がある。また、デジカメの普及により、文書に写真を添付することが簡単に出来るようになった。また、画像の形式にはいろいろな種類がある。
- (4) 映像：デジタル化と圧縮技術およびネットワークの高速化により、インターネットで映像を伝達することも簡単に出来るようになった。
- (5) 音楽・音声：音をデジタル化する技術により、CDやMDが生まれた。また、圧縮技術により、インターネットで最新の音楽を購入出来るようになった。

2. プрезентーション

プレゼンテーションとは直訳すると提示とか発表といった意味になる。一般にプレゼンテーションの目的は、限られた時間の中で明確な目的を持って情報を相手に伝えることにより、相手の理解と納得を得ることである。ビジネスの世界では、これに加えて、自分の意図する方向へ相手の意志を導くことにより、商品を購入させたり企画を実現させることである。

プレゼンテーションを行うときは、専用のソフトウェアを使うことにより、マルチメディアを使った有効なプレゼンテーションを行うことが出来る。

□例題6.4□ 次の文章の①～⑩に当てはまる最も適当な語句を選択群から選び、記号で答えなさい。(第32回)

マルチメディアは情報をデジタル化するだけではなく、データを圧縮することによってデータを小さくすることにより、伝達や保存が容易になる。また、圧縮されたデータを利用できる元の状態に戻すことを①という。①により、完全に元の情報に戻せる圧縮を②圧縮といい、完全に元には戻らず情報が若干失われる圧縮を③圧縮という。画像の場合、②圧縮の例としては④、③圧縮の例⑤としてあげられる。

映像の場合は、ハードウェアを使って圧縮するほうが⑥に負荷をかけず高速に行えることが多い。映像の圧縮には⑦方式がよく使われる。

また、音楽や音声の圧縮では、拡張子が「.WAV」のファイル形式が標準で用いられているが、これは音声を⑧録音した形式である。また、実際の音ではなく、楽器や楽譜などの演奏情報をファイル化したものを作成するファイルと呼び、⑩などで音を再現できる。

解答群

- | | | | | | | |
|---------|-----------|---------|--------|--------|---------|--------|
| ア. MPEG | イ. JPEG | ウ. MIDI | エ. BMP | オ. PCM | カ. TIFF | キ. CPU |
| ク. アンプ | ケ. シンセサイザ | コ. 伸張 | サ. 非可逆 | シ. 可逆 | | |

解説

圧縮したデータを元に戻す場合、ソフトウェアでは解凍という言葉がよく使われるが、正しくは「伸張」である。画像ファイルの種類が多いが、BMP形式は非圧縮である。圧縮形式には、JPEG、GIF、TIFF、PNGなどがある。映像の種類としては、MPEGが一般的であるが、この他にAVI、QuickTime、モーションJPEG、リアルビデオなど多くの形式がある。

音声の形式としては、PCM録音によるWAVファイルと、音楽情報のMIDIファイルが主流である。最近では、不要な音域を削除することにより圧縮率を高めたMP3などの圧縮型PCM形式もある。

解答

①コ ②シ ③サ ④カ ⑤イ ⑥キ ⑦ア ⑧オ ⑨ウ ⑩ケ

練習問題

15 マルチメディアに関する次の記述中の①～⑤に入れるべき適当なものを解答群の中から選び、記号で答えなさい。(第28回)

- (1) パーソナルコンピュータと などの電子楽器を 規格のインターフェースを介して接続し、コンピュータからの制御によって演奏させたり、鍵盤などからの入力データをコンピュータに保存し、それをさまざまに活用することができる。
- (2) 一般に画像データは大きいので、デジタルカメラの画像データは されて保存されている。ファイル形式としては 形式が一般的に用いられるが、 を高くするとファイルサイズは小さくなるが画像が荒くなる。

解答群

ア. MPEG	イ. JPEG	ウ. MIDI	エ. BMP	オ. PCM	カ. 圧縮
キ. 伸張	ク. 圧縮率	ケ. 輝度	コ. シンセサイザ		

16 プрезентーションに関する次の記述中の①～⑩に入れるべき適当なものを解答群の中から選び、記号で答えなさい。(第33回)

プレゼンテーションをするためには、まず、プレゼンテーションをする相手の専門性、、必要性、経験などをすることによりを設定し、言い落としを防ぎ、相手を納得させるための材料を用意する。実際のプレゼンテーションでは、ポイント、構成、およびを考慮するとともに、話法、、プレゼンテーション機器の使い方なども大切な要素となる。

話し方の基本としては、の使い方、語彙、言葉のとイントネーションについて留意するとともに、話すときのにも注意する必要がある。プレゼンテーション機器の選択については、情報の視覚化に考慮して、スライド、、VTRなどを用いると効果的である。最近では専用のプレゼンテーションソフトがあるので、パソコンを提示用のにつないで用いることが多い。

解答群

ア. 興味	イ. マナー	ウ. プロジェクタ	エ. 分析	オ. 時間配分
カ. アクセント	キ. OHP	ク. 敬語	ケ. 目標	コ. 表情

17 次の文章中の①～⑩に入れるべき適当なものを解答群の中から選び、記号で答えなさい。

人間の耳に聞こえる音声は20Hz～kHzの空気の振動であり、これをマイクロフォンなどを用いてに変換する。この信号は信号であり、これをコンピュータで扱うことのできる信号に変換することを変換という。

この作業は、、、の順に3段階の処理を行う。1秒間に何回データを取り出すかを周波数といい、音楽用CDでは一般にkHzとなっている。また、そのデータをどのくらいの細かさに分けるかをといい、CDでは最終的にビットの2進数に置き換えられるので、65536段階に分割される。

解答群

ア. 量子化	イ. 標本化	ウ. 符号化	エ. ディジタル	オ. アナログ
カ. D/A	キ. A/D	ク. 電気信号	ケ. 8	コ. 15 サ. 16
シ. 20	ス. 22.1	セ. 32	ソ. 44.1	タ. 65536

6.5 情報化の課題

要 点

学校や社会で守るべきエチケットがあるように、インターネットの世界でもネットと呼ばれる守るべきエチケットが存在する。また、簡単に情報を得たり発信出来たりするために、知らず知らずのうちに他人の著作権を侵害している可能性もある。あるいは、他人のプライバシーを傷つけたり、自分のプライバシーが侵害されたりといった問題も生じる。さらに、長時間コンピュータの前に座ることが多くなるため、健康上の問題が生じることがある。

このように、情報化が進む中で、私たちが知っておかなければならぬ知識が多くある。

□例題6.5□ 次の①～⑩に当てはまる最も適当な語句を解答群から選び、記号で答えなさい。(第30回)

コンピュータを使用する環境は、それを扱う人間にとって快適で好ましい環境でなければならない。長時間ディスプレイを見続けることは①や肩こり、頭痛などの②の原因となる。特に、画面の③や反射光のまぶしさはその大きな原因なので、天井などの光源が直接ディスプレイに写り込むのを防ぐために、蛍光灯などの光源に④と呼れるカバーを付け、直接光源の光が見えないようにしたり、ディスプレイに⑤を付けたり、⑥処理を施したディスプレイを使うなどの配慮が必要である。また、長時間のタイピングにより、⑦が起きたり、座ったままの姿勢を続けることが⑧の原因になることがある。また、オフィスの騒音も頭痛やいらいらの原因になるので、BGMを流すなどして騒音を⑨する方法も採られる。この他、コンピュータに過度に依存するなどの精神的、心理的な障害を総称して⑩と呼ぶことがある。

解答群

- | | | | | |
|----------|----------|----------|------------|---------|
| ア. ちらつき | イ. VDT障害 | ウ. マスキング | エ. テクノストレス | オ. 腱鞘炎 |
| カ. ドライアイ | キ. 腰痛 | ク. ノングレア | ケ. ルーバー | コ. フィルタ |

解説

腱鞘炎を防ぐために、テクニカルタッチのキーボードが考案されたり、CRTディスプレイのかわりに液晶ディスプレイが使われるようになってきた。また、ディスプレイの反射はほとんど無くなつた。このように、環境面では改善が見られるが、テクノ依存症やテクノ恐怖症などのテクノストレスはまだ注意が必要である。

解答

- ①カ ②イ ③ア ④ケ ⑤コ ⑥ク ⑦オ ⑧キ ⑨ウ ⑩エ

練習問題

18 次の記述中の①～⑧に入れるべき適当なものを解答群の中から選び、記号で答えなさい。(第28回)

知的所有権には、大きく分けて、新しい発明の保護のための①などの②と、表現形式を保護する著作権の二つがある。また、著作権には、③や氏名表示権などの著作者人格権と、④、頒布権などの著作者財産権がある。

また、記号やシンボルを保護する法律もあり、トレードマークは日本では⑤と呼ばれ保護されている。また、ホームページに芸能人の顔写真を無断で掲載した場合、⑥を侵害することになる。

さらに、クラシックなどの既に著作権が失われた昔の曲であっても、その演奏をインターネットで無断で流すと、演奏者の⑦に違反することになるので注意が必要である。また、インターネットでダウンロード出来る⑧と呼ばれるソフトウェアは、使用するのは無料であるが、著作権は存在しているので誤解してはいけない。

解答群

- | | | | | |
|--------------|-----------|--------|----------|--------|
| ア. 肖像権 | イ. 著作隣接権 | ウ. 複製権 | エ. 公表権 | オ. 特許権 |
| カ. フリーソフトウェア | キ. シェアウェア | ク. 商標 | ケ. 産業財産権 | |

19 次の記述中の①～⑤に入れるべき適当なものを解答群の中から選び、記号で答えなさい。(第31回)

情報を伝送するとき、情報が第三者に漏れないようにするために、情報を一定のルールに従って変換することを①といい、このとき使うパラメータを①鍵という。また、これをもとのデータに戻すことを②といい、このとき使うパラメータを②鍵という。これら二つの鍵が同じ場合を③とよぶ。また、①の方法としては、文字を一定の規則で他の文字に置き換える④と、一定の規則で文字を並べ換える⑤がある。

解答群

- | | | | | |
|------------|------------|--------|--------|--------|
| ア. 換字式 | イ. 転置式 | ウ. 復号化 | エ. 暗号化 | オ. 機密化 |
| カ. 秘密鍵暗号方式 | キ. 公開鍵暗号方式 | | | |

20 次の記述中の①～⑩に入れるべき適当なものを解答群の中から選び、記号で答えなさい。

①は悪意のあるプログラムで、プログラムやデータなどの破壊を起こすことがある。①はフロッピーディスクなどの②メディアや、インターネットの③などから④することがある。インターネットで⑤したプログラムファイルを実行したり、③の添付ファイルを開いたときに④することが多い。①が活動すると、データやプログラムが破壊されることが多いので、データの⑥をとっていない場合、そのデータは完全に失われてしまう。①は、⑦を使用して駆除することが出来るがデータはほとんど修復できないことが多い。したがって、最も有効なのは①対策プログラムを常駐させて常に④を防ぐことである。

また最近では、⑧と呼ばれる悪意のあるプログラムもあり、個人の知らない間にコンピュータの情報を集めて外部に送信されることもある。

さらに、インターネットに接続する機会が増えると、コンピュータそのものに外部から攻撃を受けて、ひどいときには勝手にコンピュータが接続されて操作されてしまうことさえある。この様なことを防ぐために、インターネットと接続する場合に⑨を通して接続すれば比較的安全である。さらに、⑩と呼ばれる機能を利用すればより安全にインターネットに接続できる。

解答群

- | | | | | |
|------------|-----------|---------------|-----------|-----------|
| ア. ワクチンソフト | イ. メール | ウ. リムーバブル | エ. ルーター | オ. ダウンロード |
| カ. HDD | キ. ハブ | ク. コンピュータウィルス | ケ. 感染 | コ. スパイウェア |
| サ. 潜伏 | シ. フォーマット | ス. ファイアウォール | セ. バックアップ | |

21 次の文章に関係の深い語句を解答群の中から選び、記号で答えなさい。

- ①生体認証とも呼ばれ、人間の体の一部（生体器官）の情報を用いて行う個人認証技術。
- ②企業などの団体において、秘密情報・個人情報等の管理や、リスク管理についてまとめた規範のこと。
- ③ある特定のコンピュータネットワークとその外部との通信を制御して、内部のコンピュータネットワークの安全を維持することを目的としたソフトウェアあるいはそのソフトウェアを搭載したハードウェア。
- ④コンピュータネットワークに不正に侵入したり、システムの破壊・改竄などの悪意を持った行為を行う者。
- ⑤ユーザーに関する情報を集めて記録し、ユーザーに無断で集めた情報を予め設定された特定の企業や個人等に送信するソフトウェア。

解答群

- | | | |
|---------------|-------------|---------------|
| ア. スパイウェア | イ. ハニーポット | ウ. セキュリティポリシー |
| エ. クラッカー | オ. Digest認証 | カ. バイオメトリクス |
| キ. コンピュータウィルス | ク. ポートスキャン | ケ. ファイアウォール |

7. アルゴリズム

7. 1 並べ替え（ソート）

要 点

代表的なアルゴリズムを学習し理解することは、プログラムを作成していくうえで大切なことである。基本的なデータの並べ替え(ソート)のアルゴリズムには、選択法、交換法、挿入法などがある。また、基本的なデータ探索においては、逐次探索法、2分法などがある。これらについては、「2級情報技術検定試験標準問題集」にも説明されているが、重要なアルゴリズムなので、本書でも問題として取り上げて解説する。

□例題7.1□ 次の流れ図は、選択ソート（直接選択法）によって、 n 個のデータを昇順(小さいものから大きいものへの順)に並べ替えるアルゴリズムを示す。流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。

ただし、並べ替えの対象となるデータは配列 $d(1) \sim d(n)$ に入力されるものとする。

解説

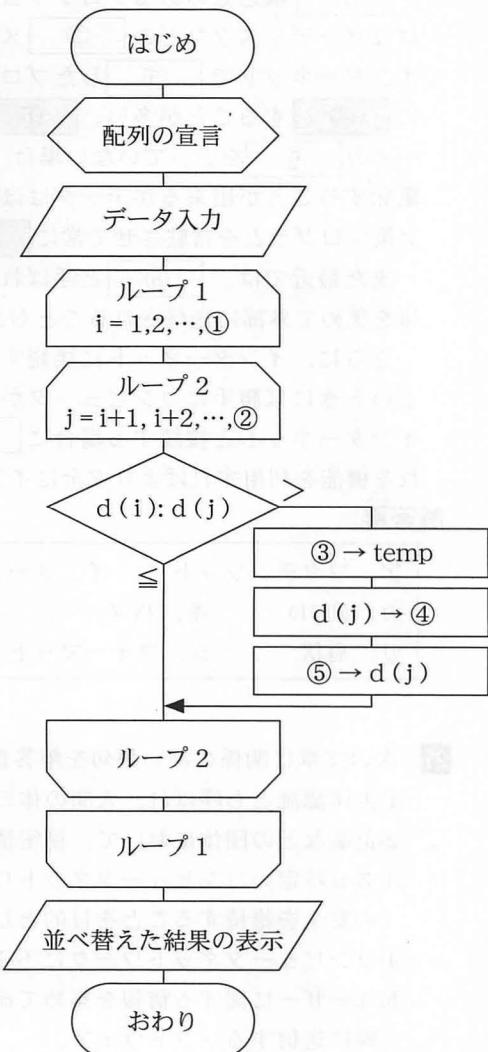
選択ソートは直接選択法とも呼ばれる。選択ソートで昇順（小さいもの順）に並べ替える場合は、まずデータ全体から最小のデータを見つけて、それを先頭のデータに入れ替える。つぎに2つ目のデータから最後までを調べてその中の最小のものを見つけて、2つ目のデータと入れ替える。これを繰り返して、データを並べ替える方法である。実際のプログラミングでは、最小値を見つけるかわりに、先頭データと2つ目以降のデータを順次比べ、そのデータが小さければ先頭データに入れ替える方法をとることが多い。この方法での並べ替えの例を下に示す。

また、データの入れ替えは、tempという作業用の変数を用意して、以下のように操作することにより入れ替える。

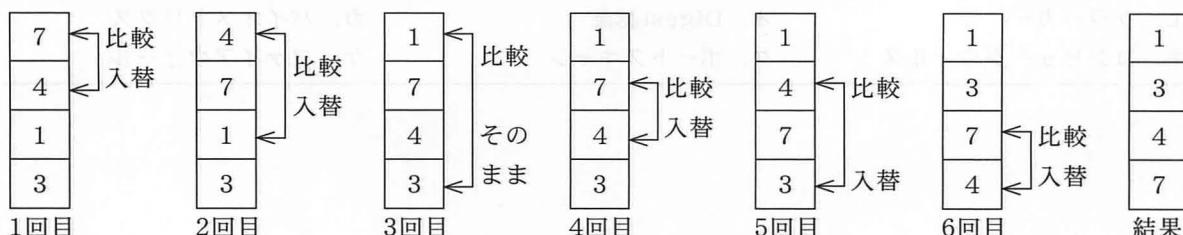
- (1) $d(i)$ の内容を変数 temp に退避する。
- (2) $d(j)$ の内容を $d(i)$ に代入する。
- (3) 変数 temp の値を $d(j)$ に代入する。

解答

- ① $n - 1$ ② n ③ $d(i)$ ④ $d(i)$ ⑤ temp



選択ソートで昇順に並べ替える例



練習問題

- 1 次の流れ図は、バブルソート（隣接交換法）によって、 n 個のデータを昇順(小さいものから大きいものへの順)に並べ替えるアルゴリズムを示す。流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。

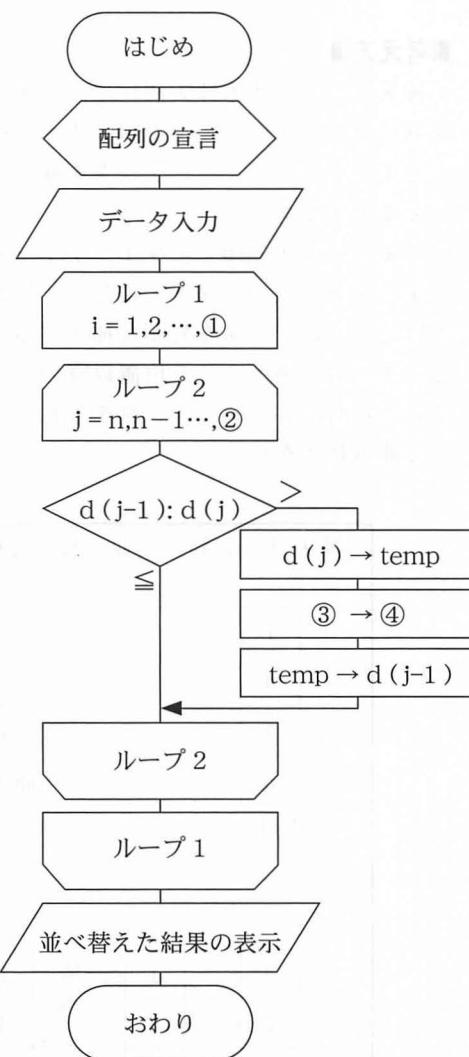
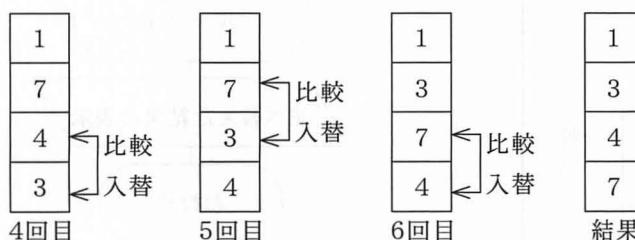
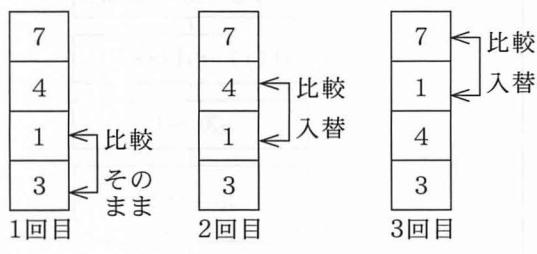
ただし、並べ替えの対象となるデータは配列 $d(1) \sim d(n)$ に入力されるものとする。

考え方

バブルソートは、隣接交換法とも呼ばれる並べ替えの方法である。昇順（小さいものから大きいものへの順）に並べ替える場合、下から順に隣り合う 2 つのデータを比較し、上のデータが大きければ入れ替えるという作業を繰り返して、データを並べ替える方法である。これは小さいデータが泡（バブル）のように上に上がっていく様子に似ているので、バブルソートと呼ばれる。この方法での並べ替えの例を下に示す。

また、並べ替えで、大きいものから小さいものへの順に並べ替えることを降順に並べ替えるという。

バブルソートで昇順に並べ替える例



補足

バブルソートは、泡（バブル）が水底から水面に向かって上がってくるように、データの最後から最初に向かって比較するのが基本である。しかし、隣接交換法という名のとおり、隣り合うデータを比較して入れ替えながら並べ替える方法であるので、データの最初から最後に向かって比較してもかまわない。この場合もバブルソートと呼ばれる。

2 次の流れ図は、挿入ソート（基本挿入法）によって、 n 個のデータを昇順(小さいものから大きいものへの順)に並べ替えるアルゴリズムを示す。流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。

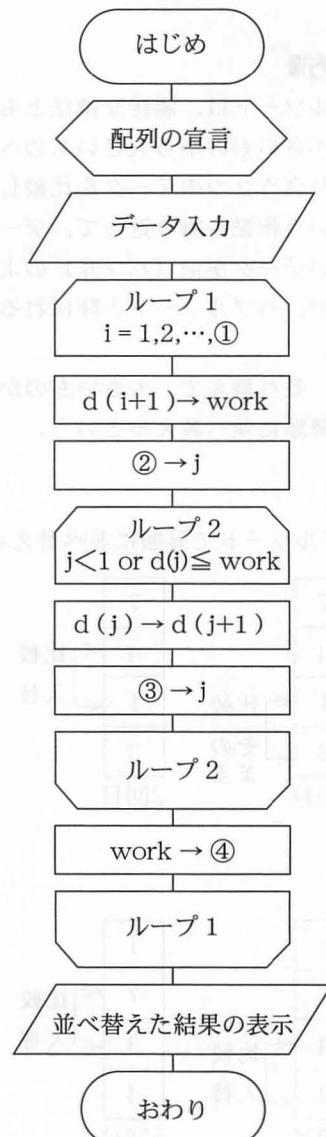
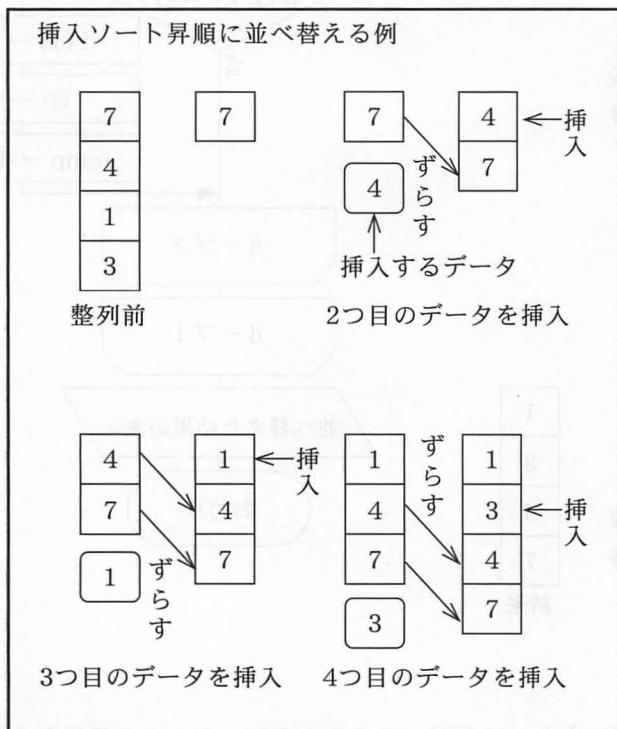
ただし、並べ替えの対象となるデータは配列 $d(1) \sim d(n)$ に入力されるものとする。

また、ループの条件は終了条件である。（第34回、第36回）

■考え方■

挿入ソート（基本挿入法）による並べ替えは、既に整列しているデータ列に対し、次のデータを挿入すべき位置を探し、挿入位置以降のデータを1つずらしてそこにデータを挿入するという操作を繰り返して並べ替える方法である。

つまり、まず先頭データ1つがソート済みデータ列であるとして、次のデータを先頭データと比較して、先頭データが大きければ1つ後ろに下げる、そのデータを先頭に挿入する。つぎに、3つめのデータのはいるべき位置を調べ、その位置以降のデータをすべて1つずつずらして、空いた場所に3つめのデータを入れる。これを最後のデータまで繰り返して並べ替えを行う。



7. 2 探索

□例題7.2□

2分探索法のアルゴリズムを示す次の流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。

ただし、探索の対象となるデータは、配列 $d(1) \sim d(n)$ に格納されており、昇順（小さいものから大きいものへの順）に並んでいるものとする。また、探索対象データは変数 key に入力されるものとする。データが見つかったときはその配列の番号を出力し、見つからなかったときは「見つからない！」と表示することとする。

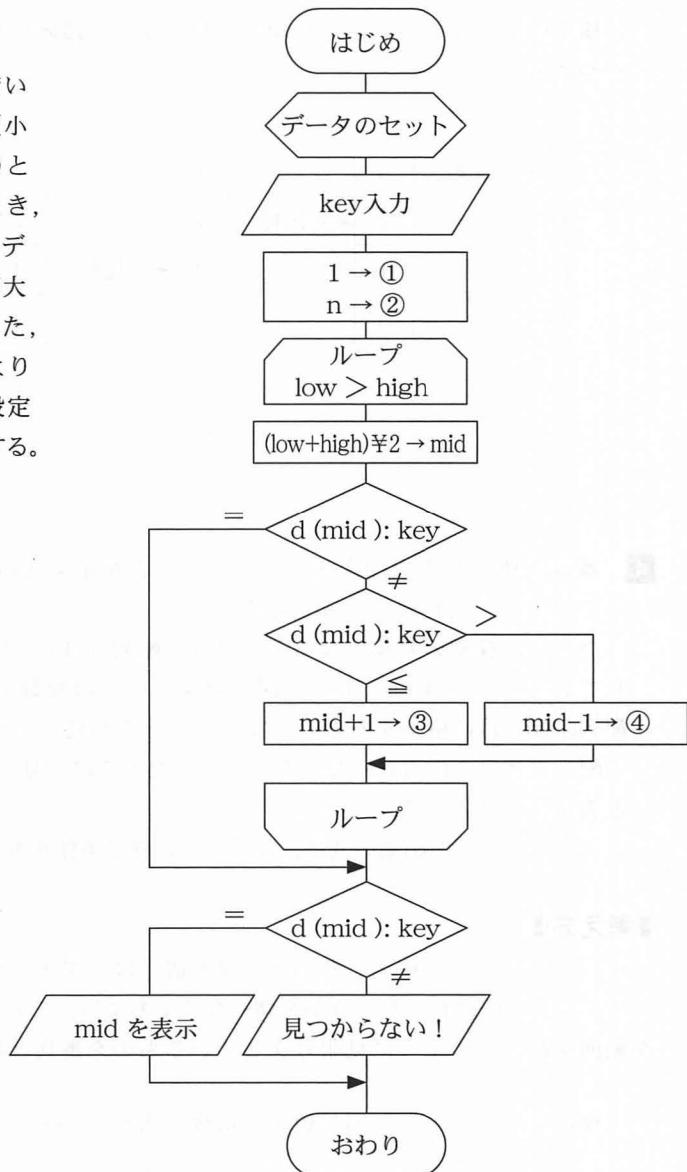
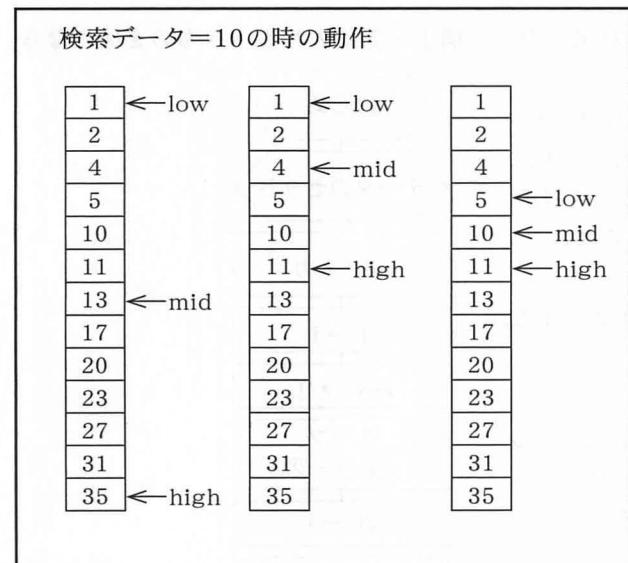
(第32回類題、第36回)

注意：流れ図中の「 $A \div B$ 」は、 A を B で割ったときの商を求める演算を表す。

また、ループ開始端の式は、繰り返しの終了条件を表すものとする。

解説

2分探索法は、データがソートされて順番に並んでいるときに有効な探索法である。いま、データが昇順（小さいものから大きいものへの順）に並んでいるものとする。探索範囲の上限を $high$ 、下限を low とするとき、 $mid = (low + high) / 2$ （整数値）の位置のデータとキーデータ（探索したい値）を比較する。キーデータの方が大きければ、探索したい値は mid より下にはない。また、キーデータの方が小さければ、探索したい値は mid より上にはない。これを元に、新たな low と $high$ の値を設定して比較を繰り返す。下に、2分法による探索の例を図示する。



解答

① low ② high ③ low ④ high

捕足

この流れ図では、構造化プログラミングに基づいて、ジャンプ（GOTO文）を使わない流れ図にしてある。ループからの脱出は、C言語などではbreak文で実現できる。データが見つかったときも見つからなかったときもループの外に処理が移るので、if文などで、データが見つかったかどうかの比較を行わなければならない。

練習問題

- 3 逐次探索法のアルゴリズムを示す次の流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。

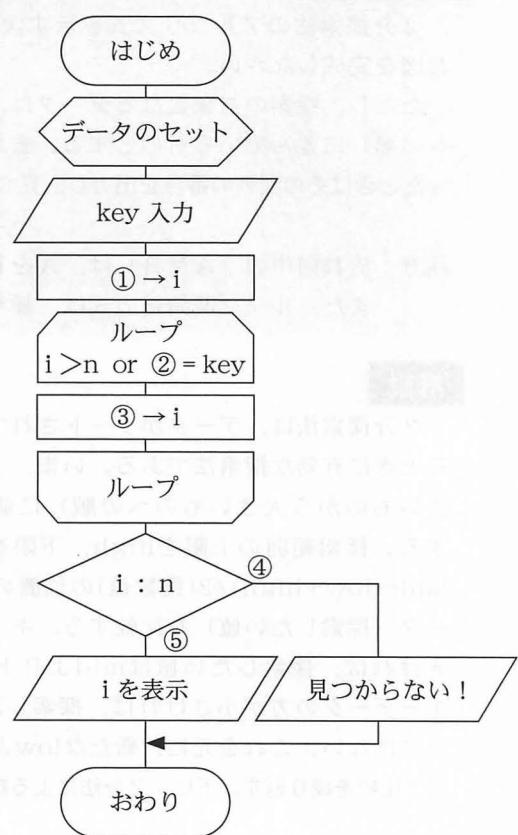
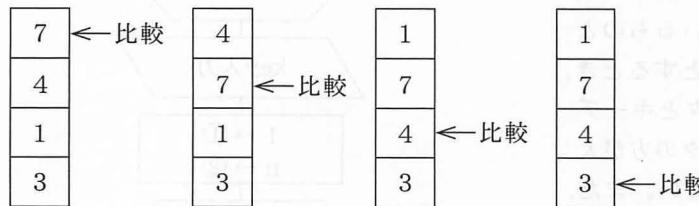
ただし、探索の対象となるデータは、配列 $d(1) \sim d(n)$ に格納されているものとする。また、探索対象データは変数 key に入力されるものとする。データが見つかったときはその配列の番号を出力し、見つからなかったときは「見つからない！」と表示することとする。

また、ループ開始端の式は、繰り返しの終了条件を表すものとする。

■考え方■

逐次探索は、データを上から順番に調べていく方法で、データが見つかった場合そこで探索を中止する。

検索データが一つしかない場合、見つかった段階で検索をやめる



- 4 番兵を使った逐次探索法のアルゴリズムを示す次の流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。

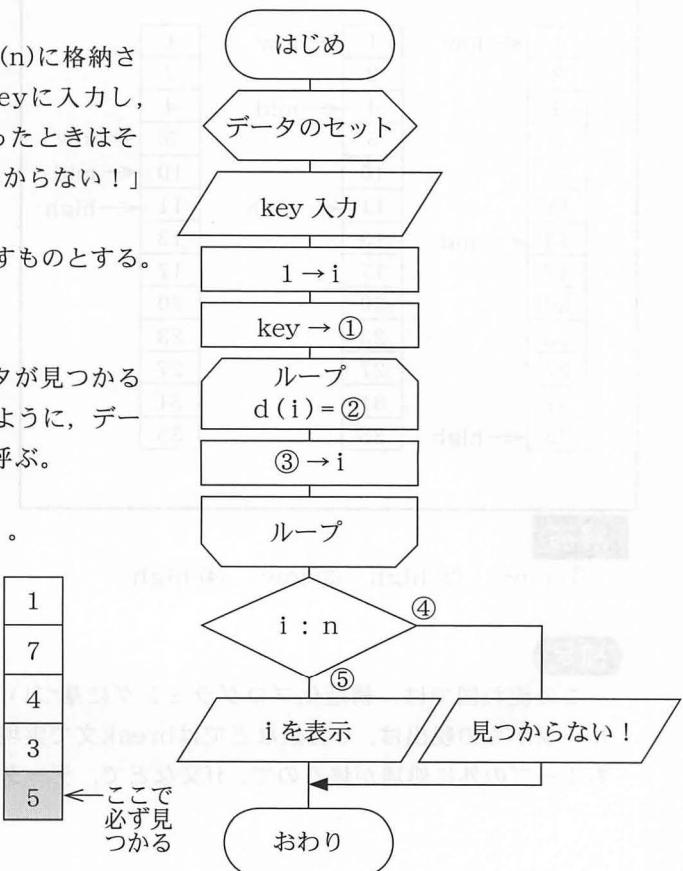
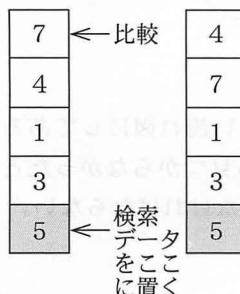
ただし、探索の対象となるデータは、配列 $d(1) \sim d(n)$ に格納されているものとする。また、探索対象データは変数 key に入力し、配列 $d(n+1)$ に格納するものとする。データが見つかったときはその配列の番号を出力し、見つからなかったときは「見つからない！」と表示することとする。

また、ループ開始端の式は、繰り返しの終了条件を表すものとする。

■考え方■

データの最後に、検索したいデータを置けば必ずデータが見つかるので、データの終わりかどうかを調べなくてもよい。このように、データ範囲を超えないように見張りをしているものを番兵と呼ぶ。

検索したいデータと同じ値を、最後に入れる（番兵）。



7. 3 順位付けと度数分布

要 点

順位付けは、人間の場合、まず最も大きい値を探しそれに1位を付け、そのつぎに大きい値を探して2位を付けて、というように行う。データの数が非常に多い場合この方法には無理がある。そこでコンピュータでは順位付けのアルゴリズムが考えられている。

また、度数分布もデータがある範囲に入っているかどうかを一つづつ調べると時間がかかるので、等間隔の度数分布の場合はうまい処理方法がある。ここでは、そのようなアルゴリズムを学習する。

□例題7.3□

次の流れ図は、配列X(1)～X(n)に入っているn個のデータの降順の順位（最大値の順位が1位で最小値が最下位）を、配列R(1)～R(n)に入れるものである。ただし、データの同じ値には同じ順位を与えるものとする。例えば、最大値100が3つあれば、それらすべての順位は1で、その次の値の順位は4となる。流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、流れ図中のループ開始端の繰り返し指定は、変数名=初期値、終値、増分である。（第24回、第33回）

解説

まず、配列R(1)～R(n)に1を代入する。

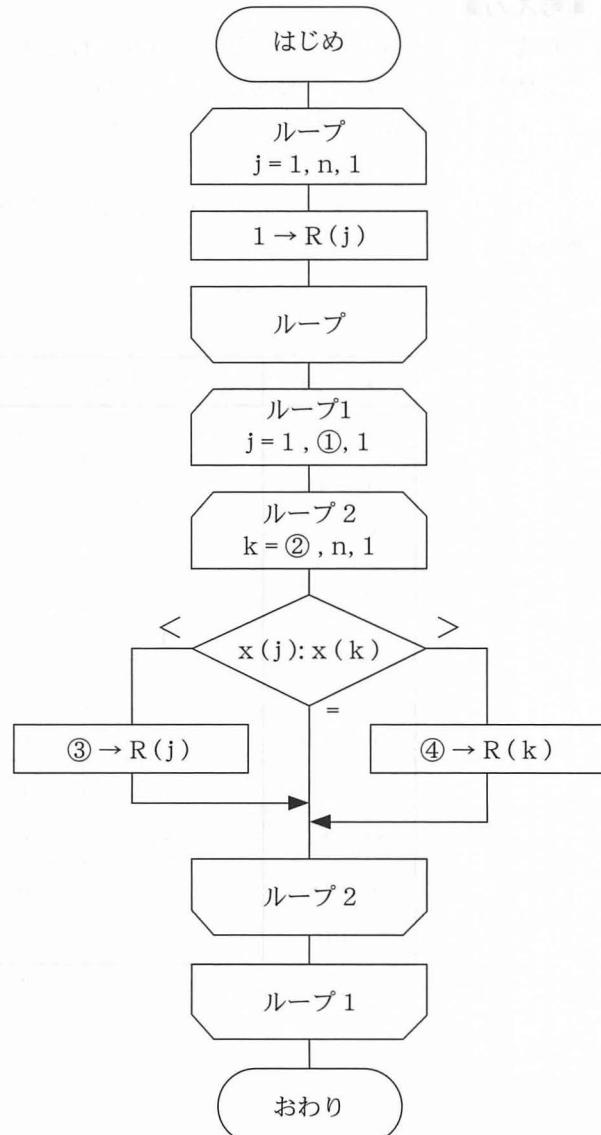
つぎに、配列X(1)の値と配列X(2)の値を比較する。このとき、X(1)の方が小さければR(1)の値を1増やし、X(2)の方が小さければR(2)の値を1増やす。つまり、小さい方の配列Rの値を1増やすことにより順位を1つ下げる。配列X(1)について配列X(n)まで比較を繰り返し、配列Rの値をセットしていく。この操作で配列X(1)の順位が決定する。

つぎに、配列X(2)について配列X(3)からX(n)を順番に比較して配列Rの値をセットしていく。この操作で、配列X(2)の順位が決定する。

同様の操作を、X(n-1)まで繰り返し行うことにより、すべての配列Xの順位が決定する。

解答

①n-1 ②j+1 ③R(j)+1 ④R(k)+1



練習問題

5 0から100点までの点数（整数）を、0～9, 10～19, …, 90～99および100の11ランクに分け、それぞれのランクの頻度を配列F(0)～F(10)に格納して度数分布を求めたい。以下の流れ図の①～④に当てはまるもっとも適当なものを解答群から選び記号で答えなさい。ただしデータの個数はn個で、0未満の数値も100を超える数値も無いものとする。（第30回類題）

ただし、解答群の工、才の「%」は演算記号で、たとえば「P % Q」は、PをQで割ったときの商（整数）を求める演算を表すものとする。

解答群

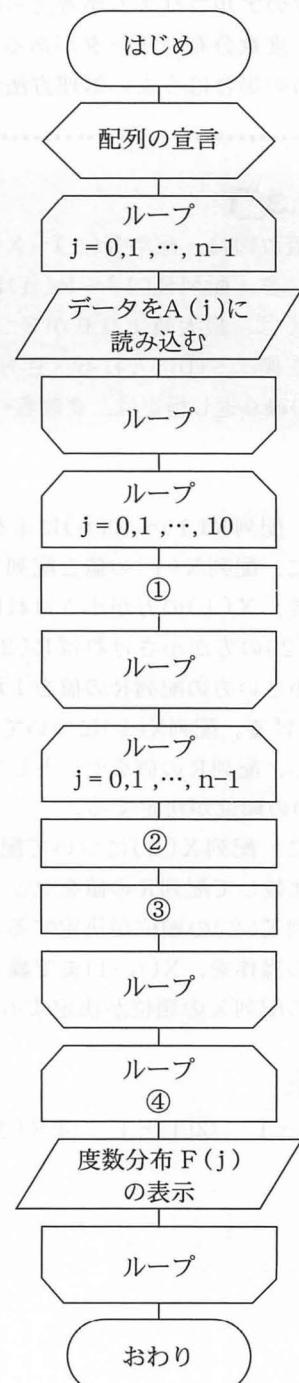
- ア. $1 \rightarrow F(j)$
- イ. $0 \rightarrow F(j)$
- ウ. $F(R) + 1 \rightarrow F(R)$
- エ. $F(j) \% 10 \rightarrow R$
- オ. $A(j) \% 10 \rightarrow R$
- カ. $j = 0, 1, \dots, n-1$
- キ. $j = 0, 1, \dots, 10$

考え方

指定された範囲のランクに分けて度数分布を調べる場合、比較を使って個数を数えるのが一般的である。しかし、この問題のように10きざみの度数を数える場合は、データの値を10で割ったときの商とランクが一致する。

これをを利用して、ランク分けを行い、度数分布を作成するのが、度数分布のアルゴリズムである。

範 囲	10で割った商	ランク
0～ 9	0	0
10～19	1	1
20～29	2	2
30～39	3	3
40～49	4	4
50～59	5	5
60～69	6	6
70～79	7	7
80～89	8	8
90～99	9	9
100	10	10



7.4 漸化式

要 点

ある規則にしたがって数や式（関数で表されることが多い）を並べたものを数列といい、それぞれの数や式は項と呼ばれる。数列で隣り合う2つまたは3つの項の間に成り立つ関係を式で表したもの漸化式という。事象によっては、漸化式で表すと簡単に結果を求めることができるものがある。漸化式は、繰り返しや再帰呼び出しと呼ばれる手法を利用すると簡単にプログラミングすることができる。

例題7.4

次の流れ図は、 n 段の階段の登り方が何通りあるかを求めて表示するものである。ただし階段の登り方は一度に1段または2段のいずれかで登るものとする。このとき、流れ図の中の空欄①～⑤に入れるべきものを解答群から選び、記号で答えなさい。ただし、階段の段数は3段以上であるものとする。また、同じ記号を2回まで使ってもよい。（第25回、第29回、第34回）

解答群

- ア. $0 \rightarrow F_2$
- イ. $1 \rightarrow F_2$
- ウ. $2 \rightarrow F_2$
- エ. \geq
- オ. $<$
- カ. $F_2 + 1$
- キ. F_3

考え方

n 段の階段の登り方を $f(n)$ とすると、

1段の登り方は、1通り

2段の登り方は、1段ずつと一度に2段の2通り

3段の登り方は、全部1段ずつ、2段1段、1段2段の3通り

4段の登り方は、1111 112 121 211 22

$$f(1)=1$$

$$f(2)=2$$

$$f(3)=3$$

$$f(4)=5$$

5段の登り方は、11111 1112 1121 1211 2111 122 212 221 $f(5)=8$

となる。

いま、第 j 段目にいるとするとき、その前は、第 $(j-1)$ 段目にいたか、第 $(j-2)$ 段目にいたかのどちらかである。

第 $(j-1)$ 段目までの登り方は、 $f(j-1)$ 通りあり、

第 $(j-2)$ 段目間での登り方は、 $f(j-2)$ 通りある。

しかも、その登り方は異なっているので、第 j 段までの登り方は、第 $(j-1)$ 段までの登り方と第 $(j-2)$ 段までの登り方を足せばよいことがわかる。

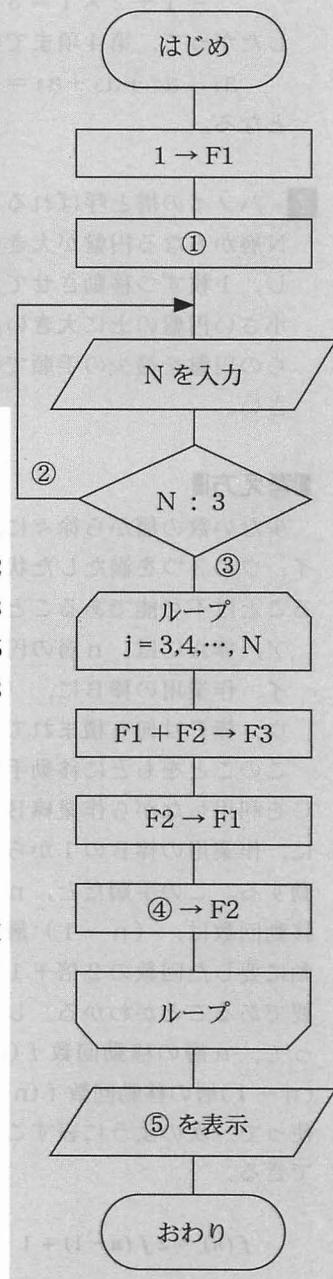
したがって、第 j 段までの登り方 $f(j)$ は、

$$f(j) = f(j-1) + f(j-2) \quad \text{ただし, } f(1)=1, f(2)=2$$

という漸化式で表すことができる。

解答

- ①ウ
- ②オ
- ③エ
- ④キ
- ⑤キ



練習問題

- 6 漸化式に関する次の流れ図の中の①～⑤に当てはまるもっとも適当な字句を図にならって記入し、流れ図を完成しなさい。（第24回、第35回）

■考え方■

右の流れ図は、漸化式が $a_{k+2} = a_{k+1} + 2a_k$ ($k \geq 1$) で表される数列の、初項から第 n 項 ($n \geq 2$) までの和を求める手順を示している。

ただし、 $a_1 = 0$, $a_2 = 1$ である。

[例] $n = 4$ の場合の例

$$a_1 = 0$$

$$a_2 = 1$$

$$a_3 = a_2 + 2a_1$$

$$= 1 + 2 \times 0 = 1$$

$$a_4 = a_3 + 2a_2$$

$$= 1 + 2 \times 1 = 3$$

したがって、第4項までの総和は、

$$a_1 + a_2 + a_3 + a_4 = 0 + 1 + 1 + 3 = 5$$

となる。

- 7 ハノイの塔と呼ばれるパズルがある。3本の棒A, B, Cがあり、棒Aには、N層からなる円盤が大きい順に積まれている。いま棒Bを作業用の棒として利用し、1枚ずつ移動させて、棒Cに移し替えることを考える。ただし、移動の際に、小さい円盤の上に大きい円盤を置くことはできないという約束ごとがある。これらの円盤を最少の手順で移し替えるときの手数を求めるアルゴリズムを完成しなさい。

■考え方■

少ない数の層から徐々に層の数を増やして実験を繰り返していくうちに、次のア、イ、ウの3つを満たした状態にならないと、棒Cの一番下に n 層の円盤を持ってくることは不可能であることに気づく。

ア. 棒Aには、 n 層の円盤だけが残っている。

イ. 作業用の棒Bに、($n - 1$)までの円盤が大きい順に積まれている。

ウ. 棒Cは何も積まれていない。

このことをもとに移動手順を整理すると、1から($n - 1$)層までの円盤を、棒Cを利用しながら作業棒Bに移動させ、残った n 層の円盤を棒Cに移動させる。次に、作業用の棒Bの1から($n - 1$)層の円盤を、棒Aを利用しながら、棒Cに移動する。この手順だと、 n 層の移動回数は、($n - 1$)層の移動を要した回数の2倍+1回必要であることがわかる。したがって、 n 層の移動回数 $f(n)$ は、($n - 1$)層の移動回数 $f(n-1)$ を使って、次のように表すことができる。

$$f(n) = 2f(n-1) + 1$$

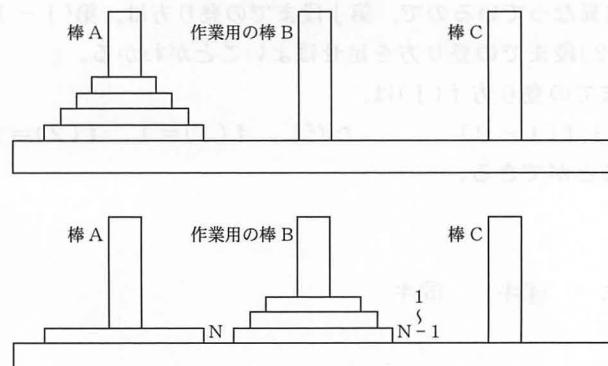
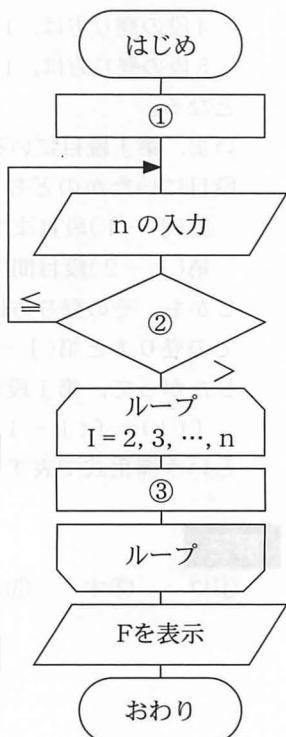
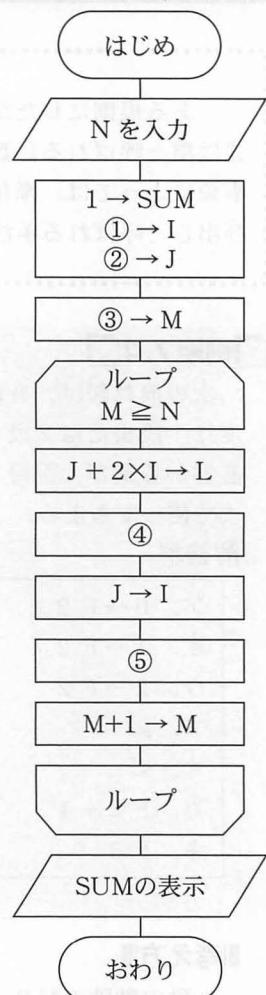


図 ハノイの塔

この漸化式を用いて流れ図を考えればよい。



7.5 簡単なアルゴリズム

要 点

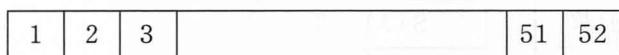
コンピュータでトランプのシャッフルと同じ作業をするにはどのようにすればよいのだろうか。従業員全員の給料を現金で支払うとき、お札や硬貨を種類ごとに何枚用意すればよいのだろうか。このように、コンピュータで効率よく処理を行うときに知っておくと便利な比較的簡単なアルゴリズムをいくつか学習しよう。

□例題7.5□

トランプのシャッフルと同様の操作をコンピュータで行いたい。52枚のトランプを1～52の数値に対応させる。A(1)～A(52)の配列に1～52の数値をランダムに入れることによりトランプを並べ替えたこととする。1～52までの乱数を発生させ、配列に代入し、すでに割り当て済みの数である場合は、まだ割り当てていない乱数が出るまで繰り返す方法が考えられるが、最後の方は条件を満たす数値が発生しにくくなり処理に時間がかかる。そこで、効率よくこの操作を行うために、次のような流れ図を考えた。空欄①～⑤に入るべきものを解答群から選び、流れ図を完成しなさい。ただし、流れ図中の「乱数(1～K)」は、1以上K以下のランダムな整数を発生させることを表す。（第35類題）

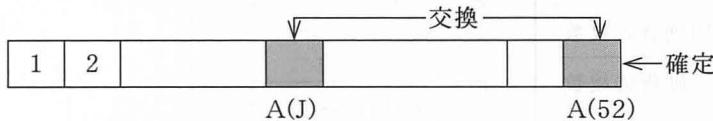
■考え方■

- ① 1～52の値を配列A(1)～A(52)に順に格納する。

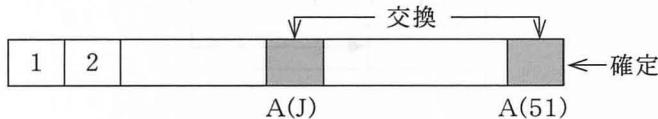


- ② 1～52の範囲の乱数Jを得て、これを添字とする配列A(J)とA(52)を交換する。

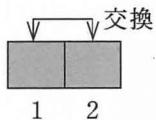
これにより、配列A(52)の値が決定するので、これを処理の対象外にする。



- ③ つぎに、1～51の範囲の乱数Jを得て、これを添字とする配列A(J)とA(51)を交換する。これにより、配列A(51)の値が決定する



- ④ 同様の処理をA(2)まで繰り返す。

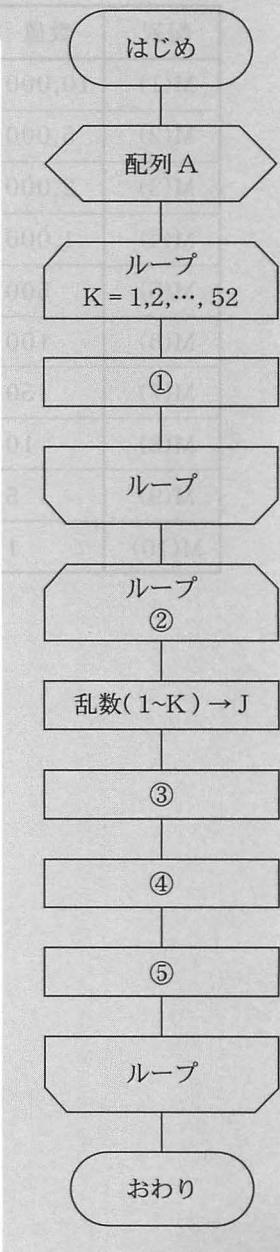


解答群

- ア. K=52, 51, …, 2
- イ. J=52, 51, …, 2
- ウ. W → A(J)
- エ. K → A(K)
- オ. A(K) → W
- カ. A(K) → A(J)
- キ. A(J) → A(K)

解答

- ① エ ② ア ③ オ ④ キ ⑤ ウ



練習問題

8 従業員100名の給料額が配列S(1)～S(100)に格納されているとき、従業員全員の給料を支払うために必要な紙幣と硬貨の数を計算して表示する流れ図を考えた。流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、紙幣は1万円、5千円、2千円、千円、硬貨は500円、100円、50円、10円、5円、1円とし、配列M(1)～M(10)に数値が格納されているものとする。また、必要な紙幣と硬貨の枚数は配列N(1)～N(10)に格納されるものとする。

ただし、流れ図中の「 \div 」は演算記号で、演算「 $P \div Q$ 」は整数 P を整数 Q で割った商(整数)を求める演算を表すものとする。また、ループ端の繰り返し指定は、変数名=初期値、終値、増分である。

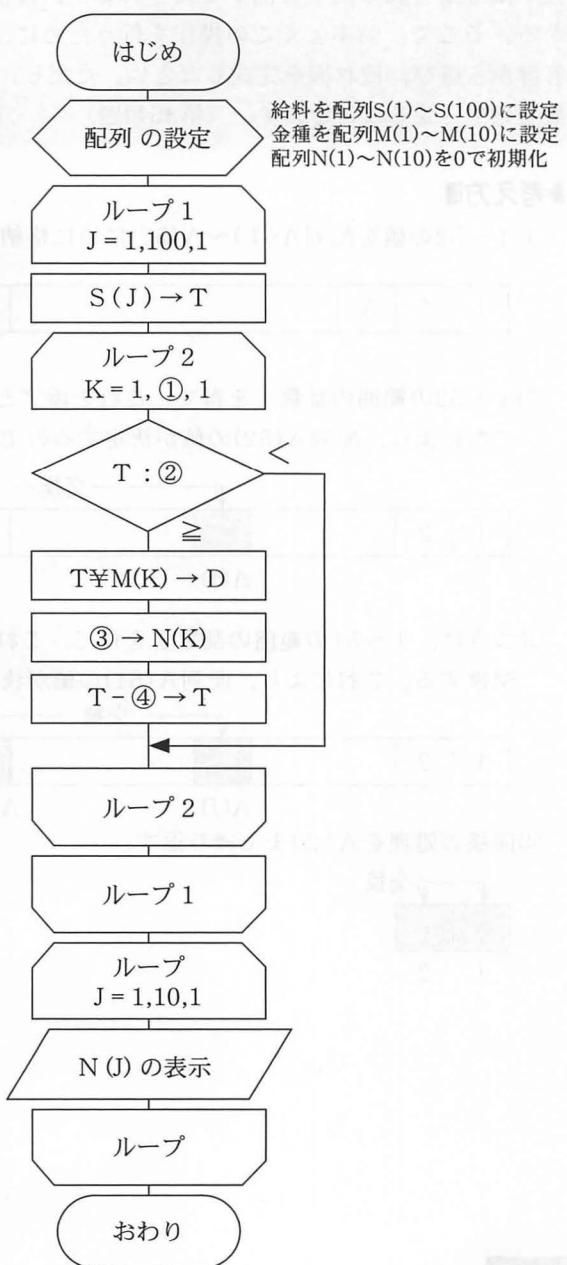
■考え方■

たとえば、ある従業員の給料が147,326円とすると、まずこれを10,000で割り、商を求めるとき14余り7,326となり、1万円札が14枚必要であることがわかる。次に147,326円から140,000円を引いた余り7,326円について、こんどは5,000円以下の必要枚数を求めていけばよい。これを従業員全員について繰り返す。

配列と格納される数値

配列	数値
M(1)	10,000
M(2)	5,000
M(3)	2,000
M(4)	1,000
M(5)	500
M(6)	100
M(7)	50
M(8)	10
M(9)	5
M(10)	1

配列	格納されるデータ
N(1)	1万円札の枚数
N(2)	5千円札の枚数
N(3)	2千円札の枚数
N(4)	千円札の枚数
N(5)	500円硬貨の枚数
N(6)	100円硬貨の枚数
N(7)	50円硬貨の枚数
N(8)	10円硬貨の枚数
N(9)	5円硬貨の枚数
N(10)	1円硬貨の枚数



- 9 次の流れ図は、入力された整数Nが連続した二つ以上の正の整数の和として表すことができるかどうか調べるものである。二つ以上の正の整数の和として表すことができる場合はその連続する整数のパターンを全て出力し、無い場合は何も出力しない。流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。（第35回）

ただし、ループ開始端1の終了値 $\frac{N}{2}$ は、切り捨てして整数とする。

また、ループ開始端2の条件は繰り返しの終了条件である。

■考え方■

整数Nが21の場合。

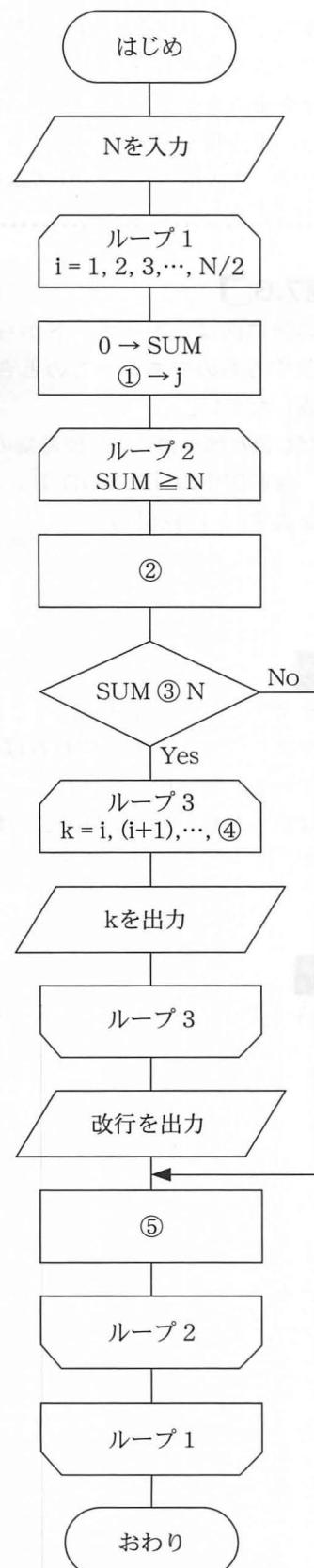
$$\begin{aligned} 21 &= 1 + 2 + 3 + 4 + 5 + 6 \\ &= 6 + 7 + 8 \\ &= 10 + 11 \end{aligned}$$

このように、3パターンで整数の和に表すことが出来る。

これら全ての場合を出力する。

また、整数Nが8の場合。

連続する数の和では表せないので、何も出力しない。



7. 6 代表的なアルゴリズム

要 点

アルゴリズムには名前の付いた代表的なアルゴリズムがある。これらのアルゴリズムを知っておくことは、プログラムを作成していくうえで大切なことである。たとえば、最大公約数を求めるユークリッドの互除法、定積分を求めるシンプソンの公式、円周率の近似値を求めるモンテカルロ法、素数をさがすエラトステネスのふるい、平方根を求めるニュートン法などはその例である。定積分や微分については後でまとめて扱うので、ここでは、代表的なアルゴリズムを学習する。

□例題7.6□

次の流れ図は、キーボードから入力されたA, B の最大公約数をユークリッドの互除法を用いて求め、結果を表示するものである。このとき、流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。

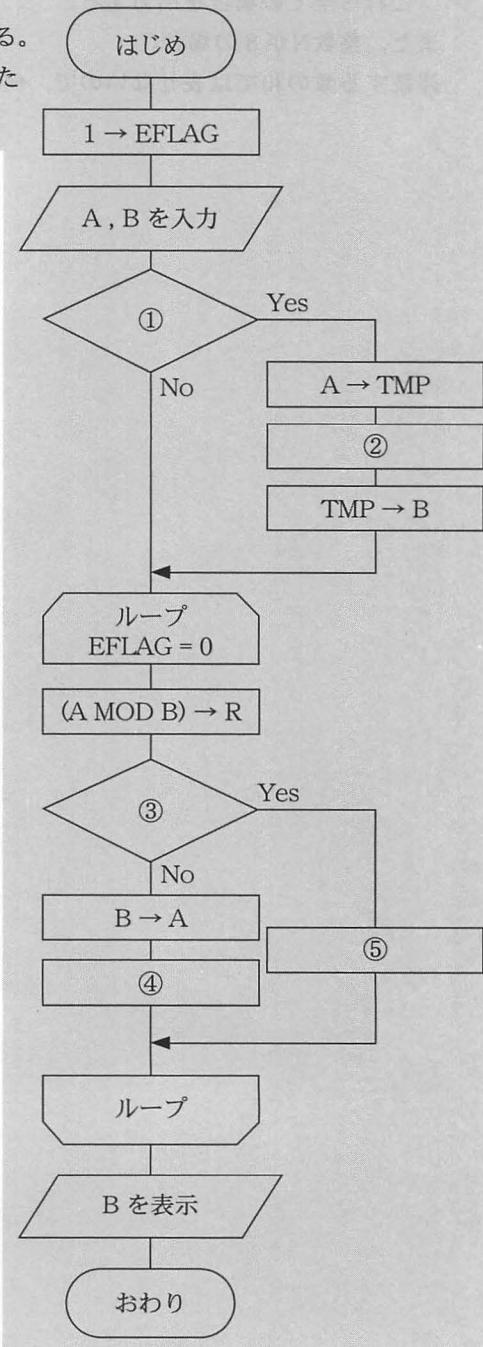
ただし流れ図中のループ開始端の中の式は、繰り返しの終了条件である。
また、流れ図中の「A MOD B」は剰余の演算を示し、AをBで割った余りを表す。（第25回）

解説

ユークリッドの互除法は、「2数のうち、大きい方の値を小さい方の値で割る。もしも割り切れれば、割った数(小さい方の数)が最大公約数である。割り切なければ、小さい方の数を余りで割り、割り切れば割った数(余り)が最大公約数である。これを割り切れるまで繰り返す。」という方法である。

解答

- ① $A < B$ ② $B \rightarrow A$ ③ $R = 0$ ④ $R \rightarrow B$ ⑤ $0 \rightarrow EFLAG$



練習問題

- 10 次の流れ図は、モンテカルロ法により円周率 π の近似値を求めて、結果を表示するものである。このとき、流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、ループ端の繰り返し指定は、変数名=初期値、終値、増分である。(第28回)

考え方

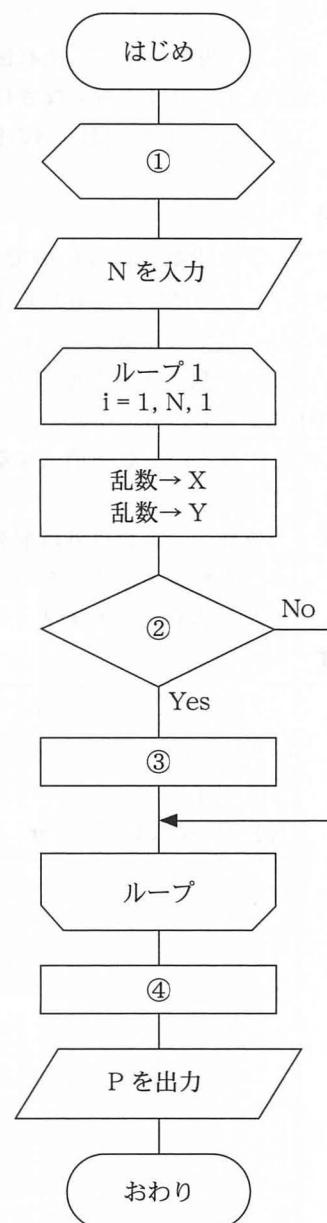
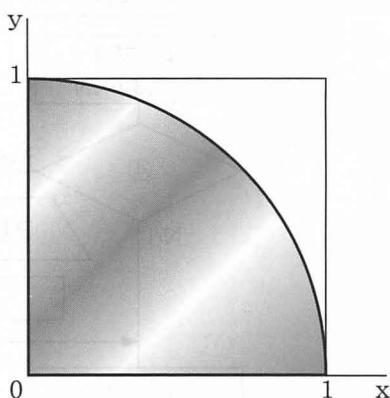
下図のように、1辺の長さが1の正方形と、正方形の頂点を中心とした半径1の円の4分の1の扇形を考える。このとき正方形の面積は $1 \times 1 = 1$ で、扇形の面積は $\pi \times 1 \times 1 / 4 = \frac{\pi}{4}$ である。したがって、正方形の中にN個の砂をランダムに落としたとき、円の中に入っている砂の数をCとすると、その数の比は面積に比例するので、Nが非常に大きな数であれば、 $\frac{C}{N} \approx \frac{\text{扇形の面積}}{\text{正方形の面積}} = \frac{\pi}{4}$ となる。この値より円周率 π の値の近似値を得ることができる。

この円周率の求め方のように確率を使ったシミュレーションによって問題を解く方法をモンテカルロ法という。十分に多くの回数シミュレーションを繰り返すことにより、近似的に解を求めることができるので、適用範囲が広い。ただし、高い精度の解を得ようとすれば計算回数が非常に大きくなってしまう弱点がある。

コンピュータで実行する場合は砂の代わりに乱数によって点を打つ。このとき、面積1の正方形の中に打たれた点が扇形の中にあるかどうかの判定は次のようにして行う。

原点を中心とする半径1の円の円周上の点の座標は $X^2 + Y^2 = 1$ であるから、あるX、Yという座標を持った点がこの扇形の内部にあるための条件は、 $X^2 + Y^2 \leq 1$ ということになる。

0以上1以下の乱数を1回に2つずつ発生させ、それらをX、Yとして、この条件を満たしていれば扇形の内部の点のカウントCを1増やす。これを十分な回数繰り返し、扇形の内部の点のカウントCを全部の点の数Nで割り $\frac{\pi}{4}$ の近似値を求める。これを4倍すれば、円周率 π の近似値Pが得られる。



練習問題

11 1と自分自身以外では割り切れない数を素数という。キーボードから2以上の整数Nを入力したとき、N以下のすべての素数と、その個数を出力するプログラムをつくりたい。

素数をさがすアルゴリズムとして、エラトステネスのふるいというアルゴリズムがあるので、これを利用する。

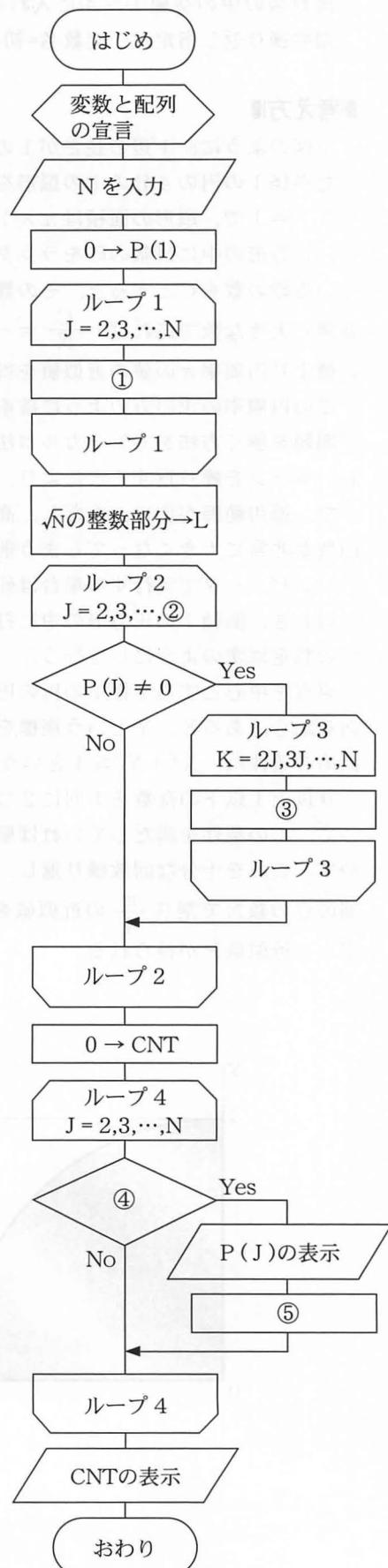
以下の、説明を読んで、流れ図の①～⑤に当てはまる語句を、解答群から選び、記号で答えなさい。ただし、流れ図中の2 J, 3 J, …, Nは、N以下のJの2の倍数、3の倍数…を表わす。（第26回）

説明

- (1) まず、1は素数ではないのでP(1)に0を格納する。その後、配列のP(J) ($2 \leq J \leq n$)にJの値を格納する。
- (2) つぎに、最小の素数2について、2の入った配列はそのままにして、2の倍数の入った配列の値を全て0にする。
- (3) 同様にJ=3, 5, 7, ……と順番に、値が0で無い配列P(J)について、数Jの倍数を消去する（0を代入する）処理を \sqrt{n} まで繰り返す。
- (4) 以上の処理で、素数以外は配列の値が全て0になったので、最後に、求めた素数を表示するため、配列の値が0でない値だけを表示する。また、素数の個数も表示する。

解答群

- | | | |
|----------------------------|-------------------------|-------------------------|
| ア. $J \rightarrow P(K)$ | イ. $J \rightarrow P(J)$ | ウ. $0 \rightarrow P(K)$ |
| エ. $0 \rightarrow P(J)$ | オ. $P(K)=0$ | カ. $P(J)=0$ |
| キ. $P(K) \neq 0$ | ク. $P(J) \neq 0$ | ケ. $K+1 \rightarrow K$ |
| コ. $CNT+1 \rightarrow CNT$ | サ. K | シ. L |
| | ス. N | |



- 12 縦・横・斜めの合計がすべて同じになるように、1から N^2 までの連続した異なる数を、正方形に並べた図形をN次の魔方陣という。いま、Nを入力したとき、N次の魔方陣を作成するプログラムをつくりたい。空欄①～⑤に入れるべきものを語群から選び、その記号を答えなさい。

ただし、流れ図中のループ開始端の繰り返し指定は、変数=初期値、終値、増分である。(第26回、第31回)

■考え方■

魔方陣は、Nが奇数の場合に限り、比較的簡単に作ることができる。
手順とその参考図を示す。

- (1) 1行目の中央に1を入れる。図(a)
- (2) 右斜め上に、次の数字を順次入れて行く。
- (3) 枠の外に出たら、反対側の端に入れる。
例：2のとき図(b)のように、4のとき図(c)のように。
- (4) すでに数が入っていて進めないときは、すぐ下に入れる。
例：6のとき図(d)のように。
ただし、これが起こるのは、入れようとしている数Jを魔方陣の次数Nで割ったときの余りが1の時のみである。
- (5) これを N^2 まで繰り返す。

N次の魔方陣を作るので、 $N \times N$ の2次元配列Aを確保する。また、魔方陣の左上の角をA(1,1)とする。(2次元配列の図)

- ヒント：a. 1行目の真ん中は、Y=1で、X=(N+1)/2である。
 b. 右上に進むには、Xを1増やし、Yを1減らせばよい。
 c. 枠の上にはみ出たかどうかは、Yが0かどうかで判断する。
 d. 右にはみ出たかどうかは、XがNより大きいかで判定する。

解答群

- ア. $N \rightarrow Y$
- イ. $X > N$
- ウ. $X < Y$
- エ. $X+1 \rightarrow X$
- オ. $Y-1 \rightarrow Y$
- カ. $Y+1 \rightarrow Y$
- キ. $Y = 0$

A(1,1)	A(1,2)	A(1,3)	A(1,4)	A(1,5)
A(2,1)	A(2,2)	A(2,3)	A(2,4)	A(2,5)
A(3,1)	A(3,2)	A(3,3)	A(3,4)	A(3,5)
A(4,1)	A(4,2)	A(4,3)	A(4,4)	A(4,5)
A(5,1)	A(5,2)	A(5,3)	A(5,4)	A(5,5)

2次元配列

		1		

(a)

		1		

(b)

		1		
4	←			4
				3
			2	

(c)

		1		
5				
4	6			3
				2

(d)

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

(e)

- 13 次の流れ図は、自然数（正の整数）Nを素因数分解するものである。次の流れ図の説明を読んで、流れ図中の空欄①～⑤に入れるべきものを解答群から選び、その記号を答えなさい。

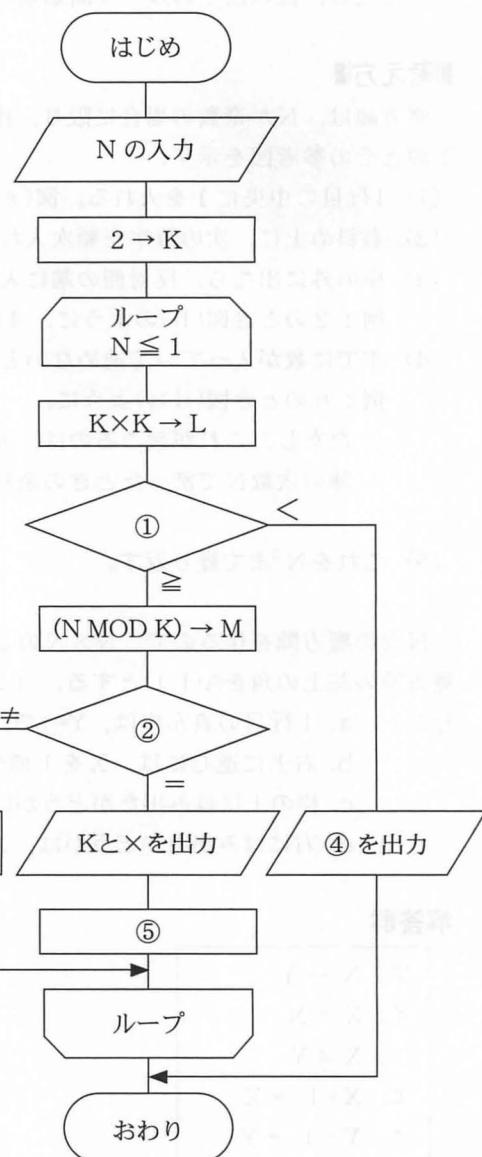
流れ図中のループ開始端の条件は、終了条件である。（第27回、第31回）

{流れ図の説明}

- (1) 素因数分解とは、自然数を素数ばかりの積の形に書き表すことをいう。1は素数ではない。
例： $60 = 2 \times 2 \times 3 \times 5$
このとき2, 3, 5は60の素因数であるという。

- (2) 流れ図中の「 $N \text{ MOD } K$ 」は 剰余の演算であり、
 N を K で割った剰余を表す。

- (3) 自然数 N が素因数の積に分解できるならば、 N は、
 $N \geq G^2$ を満たす素因数 G を持つことが数学的に証明
されている。



解答群

- | | | | | | |
|------------------------|------------------------|------------------------|-------------------------------|--------|--------|
| ア. $M : 0$ | イ. $N : 1$ | ウ. $N : L$ | エ. L | オ. N | カ. K |
| キ. $K+1 \rightarrow K$ | ク. $M/K \rightarrow K$ | ケ. $N/K \rightarrow N$ | コ. $N \times N \rightarrow L$ | | |

ア	イ	ウ	エ	オ	カ
キ	ク	ケ	コ		

ア	イ	ウ	エ	オ	カ

ア	イ	ウ	エ	オ	カ

ア	イ	ウ	エ	オ	カ
キ	ク	ケ	コ		
ス	リ	ル	ル		
セ	リ	ル	ル		
セ	リ	ル	ル		

ア	イ	ウ	エ	オ	カ

ア	イ	ウ	エ	オ	カ

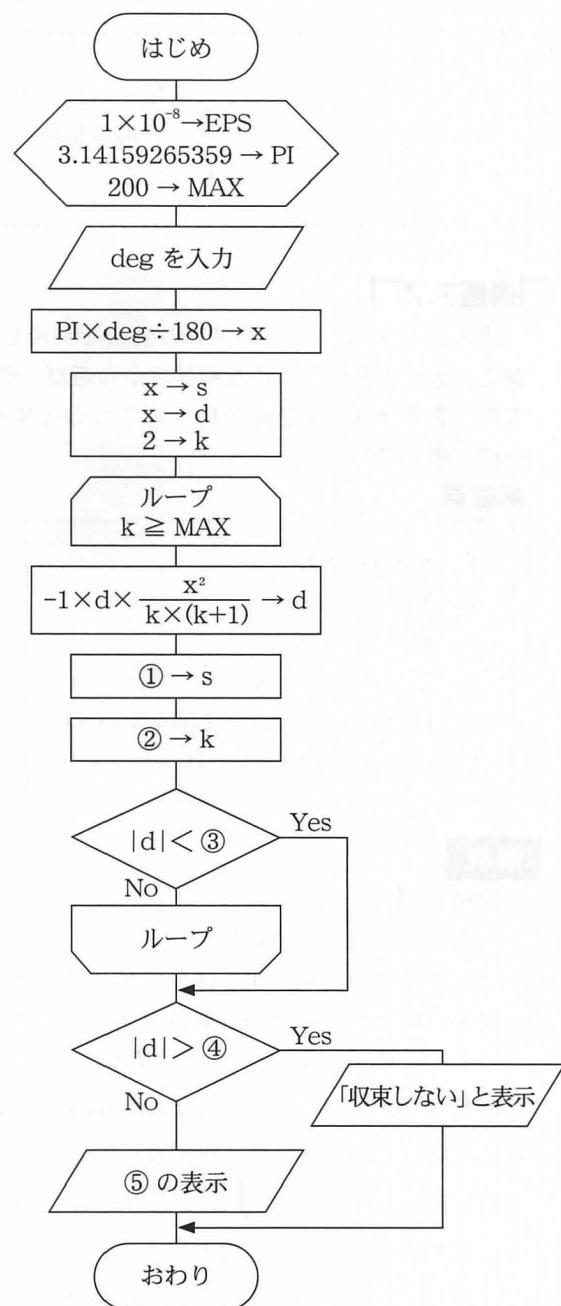
- 14 次の流れ図は、マクローリン級数展開をもつて、三角関数 $\sin x$ を求めて表示するプログラムである。空欄①～⑤を埋めて、流れ図を完成させなさい。ただし、同じ答えを2回だけ使っててもよい。また、ループ開始端の条件は終了条件である。（第31回）

■考え方■

三角関数 $\sin x$ をマクローリン級数に展開すると、次のように表される。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

増分が、あらかじめ定められた値EPSより小さくなれば計算を終了して結果を表示する。また、100回繰り返しても収束しなければ、「収束しない」と表示して終了する。



解答群

ア. k	イ. s	ウ. k + 2	エ. s + d	オ. EPS	カ. MAX	キ. S - d
------	------	----------	----------	--------	--------	----------

7.7 定積分のアルゴリズム

要 点

図形を細かく分割してその面積を求める方法を区分求積法と呼ぶ。細かく分割するときに、どのような图形に分割するかで長方形近似法、台形近似法、シンプソンの方法などの種類がある。面積を求めるアルゴリズムは定積分のアルゴリズムともいえる。

□例題7.7□

次の流れ図は、シンプソンの公式を利用して定積分 $\int_a^b f(x)dx$ の近似値を求めるものである。このとき、空欄①～⑤に入るべきものを解答群から選び、その記号で答えなさい。ただし、同じ記号を2回だけ使用してもよい。また、関数 $f(x)$ は定義されているものとする。流れ図中のループ開始端の繰り返し指定は、変数名 = 初期値、終値、増分である。

解答群

- ア. $S + D \rightarrow S$
- イ. $X + J \rightarrow X$
- ウ. $X + H \rightarrow X$
- エ. $X + 2 \times H \rightarrow X$
- オ. $H \times (L1 + 4 \times L2 + L3) \div 3 \rightarrow D$
- カ. $(B - A) \div (2 \times N) \rightarrow H$

解説

定積分 $\int_a^b f(x)dx$ の値は、下図のように $x=a$, $x=b$, $y=0$, $y=f(x)$ によって囲まれた部分の面積に等しい。

シンプソンの公式を使う区分求積法は、等分割した微小区間を二次曲線により曲線を近似して定積分の近似値を求める方法である。

いま、区間 $[a, b]$ を $2n$ 等分すると、各微小区間の幅 h は、 $h = \frac{b-a}{2n}$ となる。

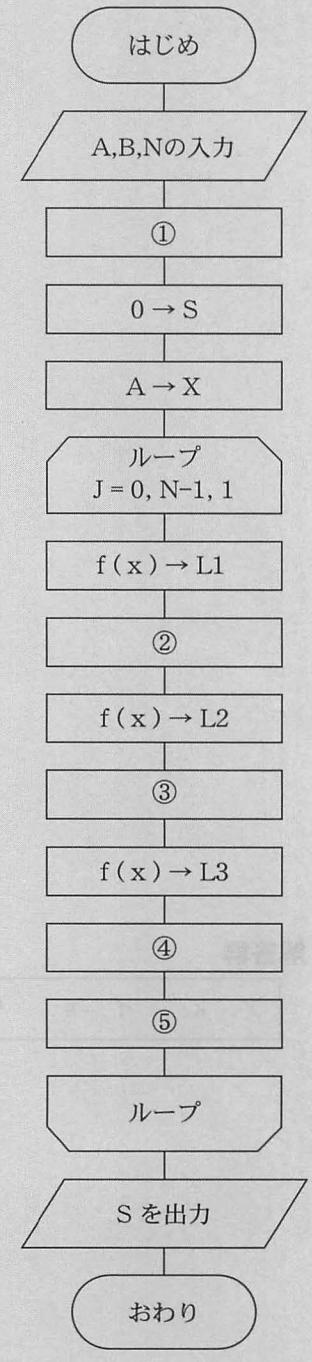
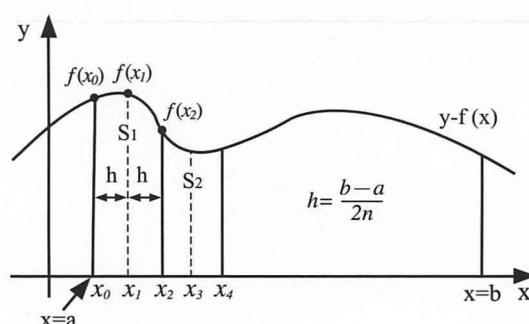
関数 $y=f(x)$ 上の3点 $f(x_0)$, $f(x_1)$, $f(x_2)$, を通る2次曲線を $y=g(x)$ とすると、微小区間 $[x_0, x_2]$ において2次曲線 $y=g(x)$ と直線 $x=x_0$, $x=x_2$ とX軸で囲まれた图形の面積 S_1 は、 $S_1 = \int_{x_0}^{x_2} g(x)dx = \frac{h}{3} \{ f(x_0) + 4f(x_1) + f(x_2) \}$ で求められる。

この式をシンプソンの公式という。

このシンプソンの公式を用いて、区間 $[a, b]$ について微小区間の面積を合計すれば定積分の値を求めることができる。

解答

- ① カ
- ② ウ
- ③ ウ
- ④ オ
- ⑤ ア



練習問題

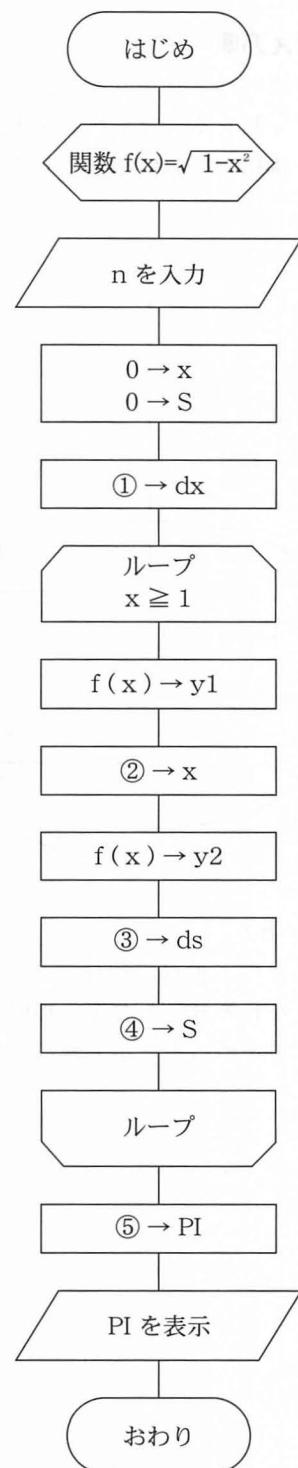
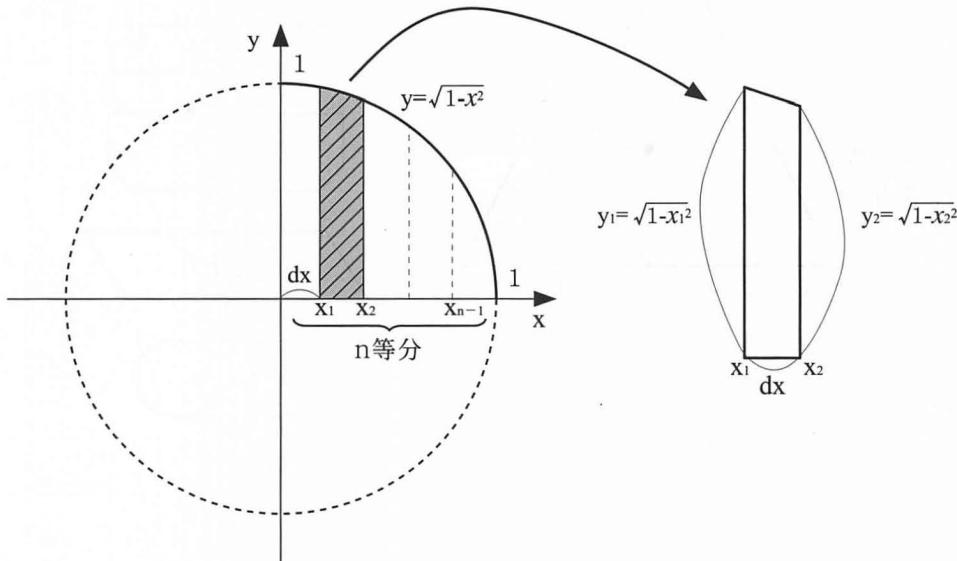
- 15 半径 1 の円の方程式は、 $x^2+y^2=1$ で表される。この式から、 $y = \sqrt{1-x^2}$ ($0 \leq x \leq 1$) で表される図形は、原点(0,0)を中心とする半径 1 の円の第 1 象限の部分であることがわかる。

したがって、定積分 $\int_0^1 \sqrt{1-x^2} dx$ は、円の第 1 象限の部分の面積を表す。

この面積を台形近似を利用した区分求積法により求め、その値をもとにして円周率 π を求めるアルゴリズムを考えた。このとき、流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、流れ図中のループ開始端の条件は、終了条件である。(第34回)

考え方

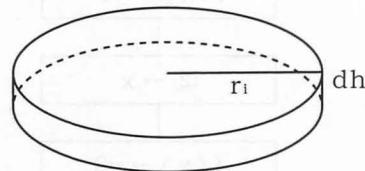
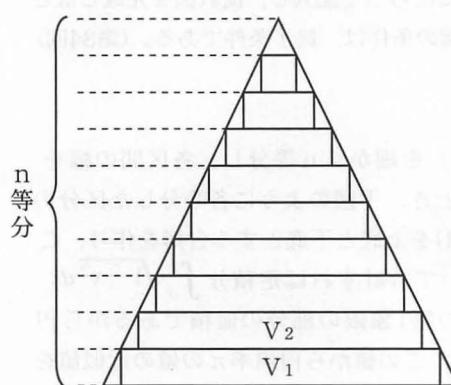
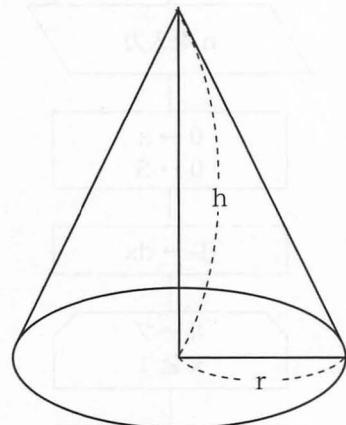
いま、上図のように、区間 $[0, 1]$ を細かく n 等分し、各区間の幅を dx とすると、 $dx = \frac{1}{n}$ となる。このとき、下図のように各等分した区分の両端の x 座標における関数の値(y 座標)を上底と下底とする台形を作り、これらの台形の面積を区間 $[0, 1]$ について合計すれば定積分 $\int_0^1 \sqrt{1-x^2} dx$ の近似値が得られる。この値は、円の第 1 象限の部分の面積であるから円の面積の 4 分の 1 である。したがって、この値から円周率 π の値の近似値を求めることができる。



- 16 次の流れ図は、底辺の半径が r 、高さが h の円錐の体積を求める処理を表している。流れ図の中の空欄①～④に入れるべきものを図にならって記入し、流れ図を完成しなさい。（第32回）

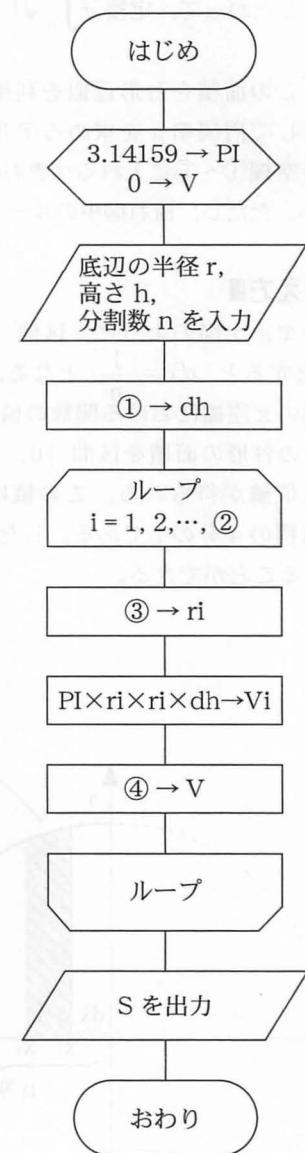
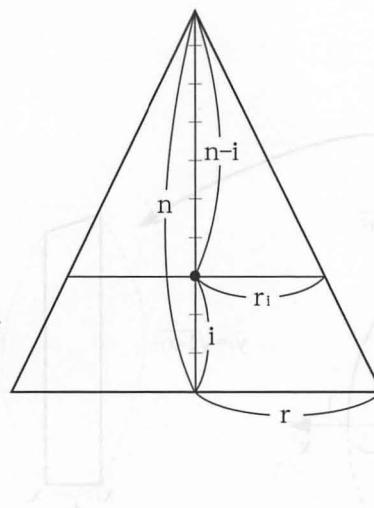
■考え方■

円錐の高さ h を n 等分し、右図のように高さ dh 、半径 ri の円柱に近似して分割する。分割数 n を十分大きな数にして円柱の体積を合計すれば円錐の体積の近似値を求めることができる。



〔ヒント〕

下から i 番目の円柱の半径 r_i は、高さ h を n 等分した右図の相似形の関係を用いて求めることができる。



7.8 根を求めるアルゴリズム

要 点

平方根や立方根を求めるアルゴリズムも数多く考えられている。ここでは代表的なニュートン法や二分法による根を求めるアルゴリズムなどについて学習する。

例題7.8

次の流れ図は、キーボードから実数A ($A > 0$) を入力し、その平方根をニュートン法により求めるものである。流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、ループ端の条件式は繰り返しの終了条件である。（第29回）

解説

$x = \sqrt{a}$ とすると、 $x^2 = a$ 、すなわち $x^2 - a = 0$ である。したがって、この問題は、 x 軸 ($y=0$) と関数 $f(x) = x^2 - a$ との交点を求めることと同じである。

ある値 x_0 における関数の接線の方程式は、

$$y - f(x_0) = f'(x_0)(x - x_0)$$

なので、 x 軸との交点を x_1 とすると、

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

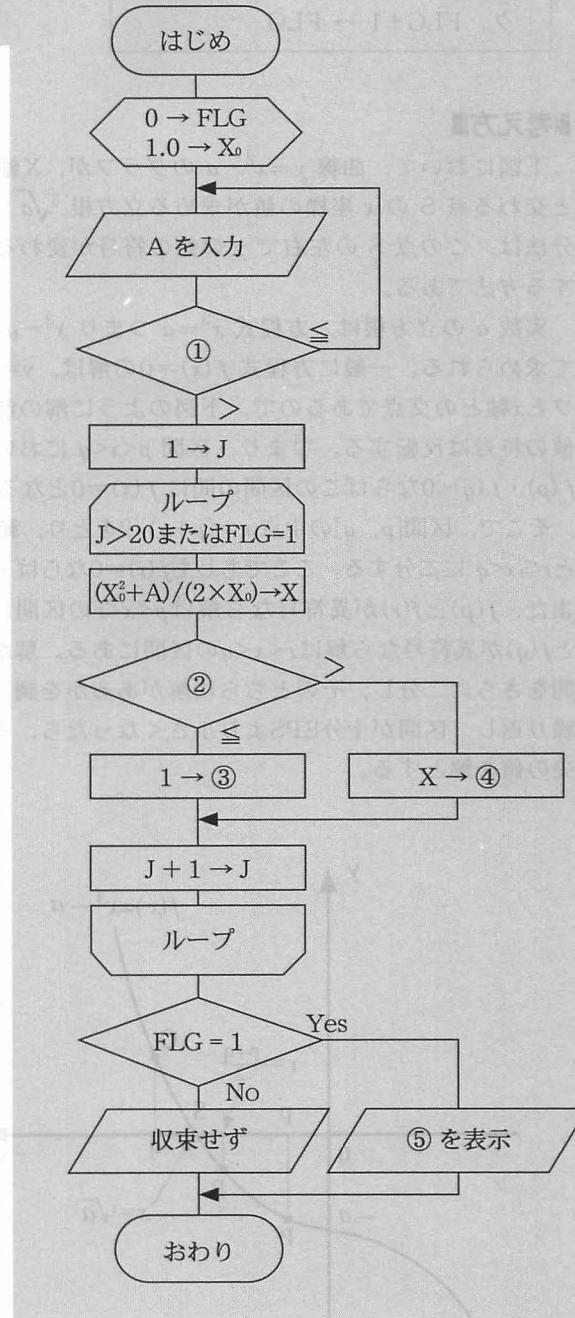
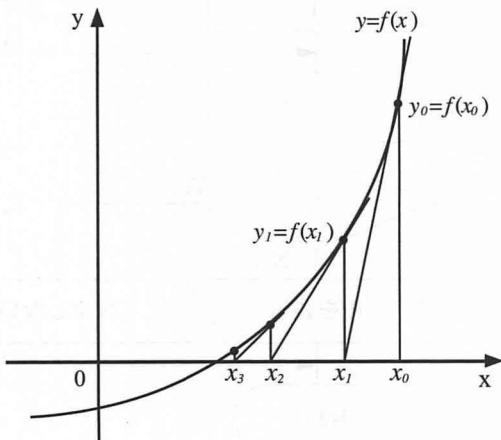
となる。したがって、漸化式は

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)} = \frac{x_j^2 + a}{2x_j}$$

となる。これにより、 x_2 、 x_3 、 x_4 と次々に値を求めることができ、 x_j と x_{j+1} の差が十分小さくなったら収束したものとし、そのときの値を求める平方根の近似値とする。

また、一定回数反復しても収束しないときは、処理を中止する。

ここでは、繰り返し回数の上限を20回とし、収束したかどうかの判定の限度を 10^{-6} とする。



解答

- ① A : 0 ② $|X - X_0| : 10^{-6}$ ③ FLG ④ X_0 ⑤ X

- 17 次の流れ図は、二分法によりある実数A(A>1)の立方根を求めるものである。このとき、空欄①～⑤に入れるべきものを解答群から選び、その記号を答えなさい。ただし、ループ端の式は、繰り返しの終了条件である。（第30回類題）

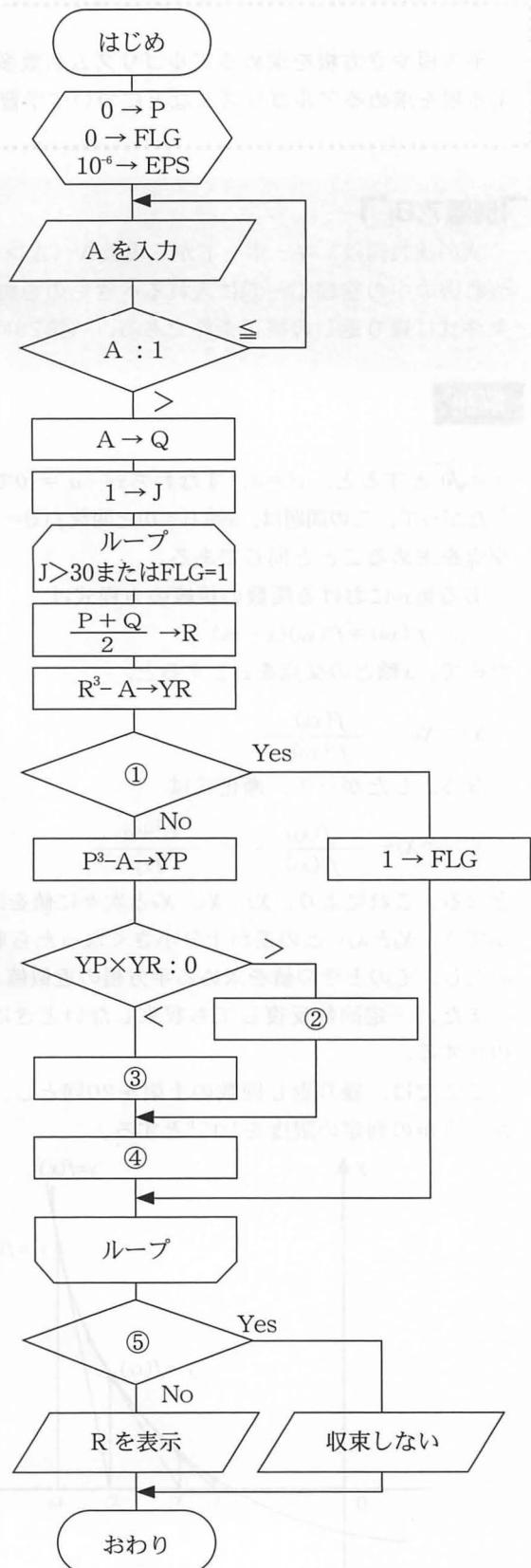
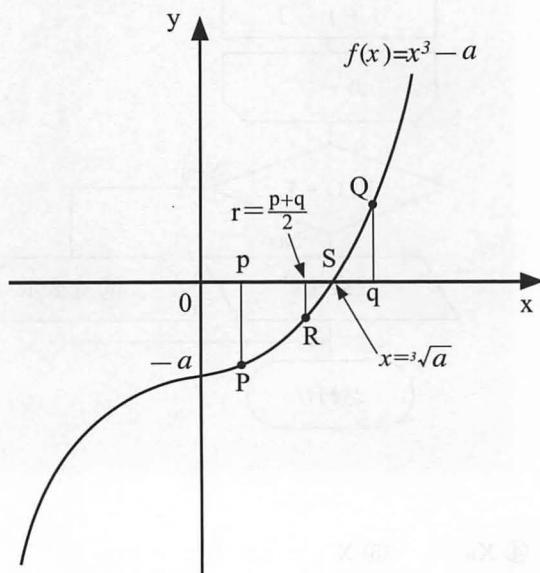
解答群

- ア. $Q - P < EPS$ または $YR = 0$
- イ. $Q - P < EPS$ かつ $YR = 0$
- ウ. $FLG = 1$
- エ. $FLG = 0$
- オ. $R \rightarrow Q$
- カ. $R \rightarrow P$
- キ. $J + 1 \rightarrow J$
- ク. $FLG + 1 \rightarrow FLG$

■考え方■

下図において、曲線 $y=x^3-a$ のグラフが、X軸の正の部分と交わる点 S の x 座標の値が求める立方根 $\sqrt[3]{a}$ である。二分法は、この点 S の左右で y の値の符号を利用することを利用する方法である。

実数 a の立方根は、方程式 $x^3=a$ つまり $x^3-a=0$ の解として求められる。一般に方程式 $f(x)=0$ の解は、 $y=f(x)$ のグラフと x 軸との交点であるので、下図のように解の前後で関数の値の符号は反転する。つまり、区間 $p < x < q$ において、
 $f(p) \cdot f(q) < 0$ ならばこの区間の間に $f(x)=0$ となる x が存在する
そこで、区間 $[p, q]$ の中点 $r=(p+q)/2$ をとり、範囲を $p < x \leq r$ と $r \leq x < q$ に二分する。ここでもしも $f(r)=0$ ならば r が解である。
また、 $f(p)$ と $f(r)$ が異符号なら解は $p < x < r$ の区間にあり、 $f(r)$ と $f(q)$ が異符号なら解は $r < x < q$ の区間にある。解がある方の区間をさらに二分し、そのどちらに解があるかを調べる。これを繰り返し、区間が十分 EPS より小さくなったら、その区間の中の値を解とする。



- 18 次の流れ図は、2の平方根の近似値を求めるものである。流れ図の中の空欄①～⑤に入れるべきものを図にならって記入し、流れ図を完成しなさい。ただし、ループ終了端の条件式は終了条件である。（第33回）

■考え方■

$\sqrt{2}$ の定義は、平方すると2になる正の数である。これを利用して、近似値の各桁を順次求めていくことができる。

最初の桁[1の位]

x	0	1	2	3	4	5	6	7	8	9
x^2	0	1	4	9	16	25	36	49	64	81

1から順番に平方して2と比べると 1^2 は2よりも小さいが 2^2 は2よりも大きいことがわかる。したがって

$$1^2 < 2 < 2^2 \\ \therefore 1 < \sqrt{2} < 2$$

であり、 $\sqrt{2}$ は1と2の間にあることがわかる。

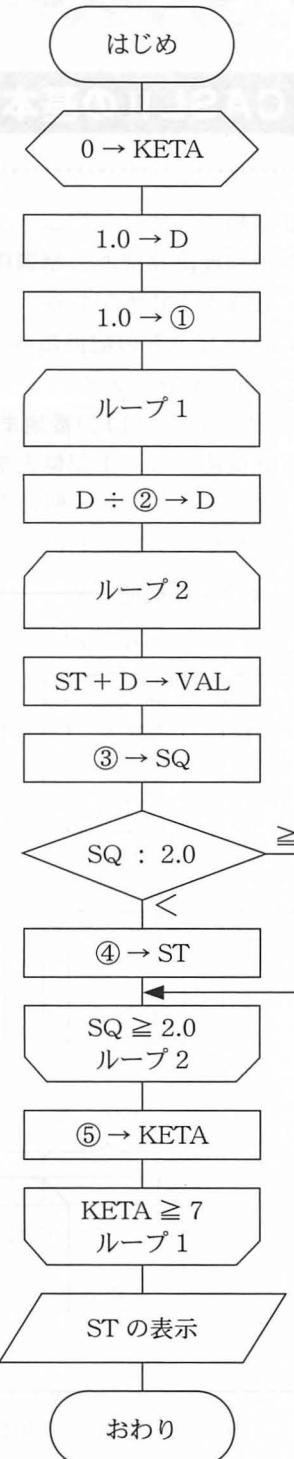
次に、小数第1位について同じことをする。

x	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
x^2	1.00	1.12	1.44	1.69	1.96	2.25	2.56	2.89	3.24	3.61

この比較から 1.4^2 は2よりも小さいが 1.5^2 は2よりも大きいことがわかる。したがって、 $\sqrt{2}$ は1.4と1.5の間にある。

同様に、小数第2位、小数第3位と行っていけば近似値を求めることができる。

右の流れ図は、 $\sqrt{2}$ が1と2の間にあることがわかっているとしている。また、できる限り無駄な比較は行わないようにしてある。また、小数第7位まで計算したら処理を終了している。



8. CASL II プログラムの作成能力

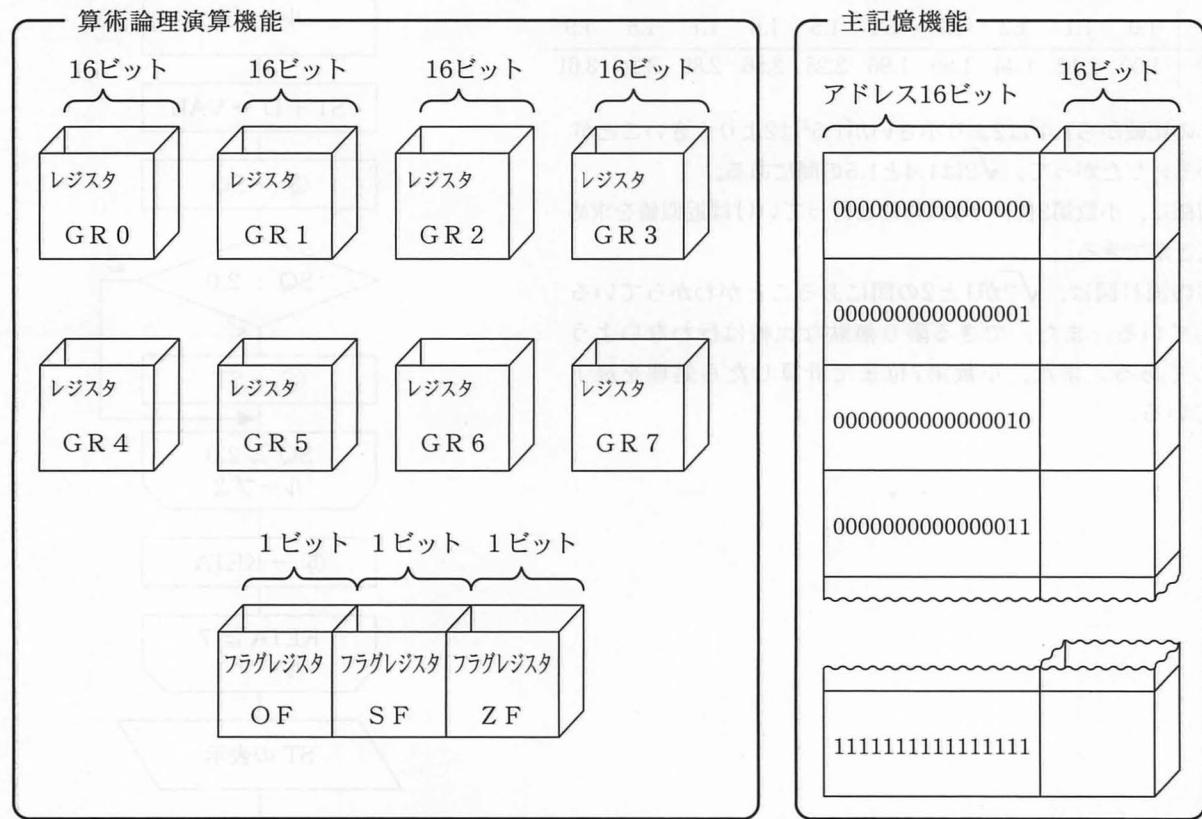
8. 1 CASL II の基本的なプログラム

要 点

CASL II が動く仮装コンピュータ COMET II も一般のコンピュータと同じく、入力機能、出力機能、主記憶機能、算術論理演算機能、制御機能をもつが、CASL II プログラミングでは、主記憶機能、算術論理演算機能を特に意識する必要がある。

主記憶は 1 語 16 ビットの記憶領域を利用して、プログラムやプログラムで使用するデータを保存するもので、その総容量は 65536 語である。また、主記憶の各語の位置を示すアドレスは 0 (0000000000000000) ~ 65535 (1111111111111111) 番地まで振られている。

算術論理演算機能では、主記憶とデータのやり取りをしたり、演算データ設定、計算結果を収容するなどの機能を持つ GR 1 ~ GR 7 の 8 個の汎用レジスタ（16 ビットの一時記憶場所）と演算命令などの結果によって、値が変化するそれぞれ 1 ビットの OF, SF, ZF の 3 つのフラグレジスタに注意を払うことが必要である。



コンピュータ内のデータは、2進数形式で扱われている。プログラムの作成・分析に当たっては、各データの意味合いをつねに考え、プログラムと接していくかなければならない。

例

0000000001000001 を $\begin{cases} \text{数値として解釈すれば } 65 \\ \text{文字として解釈すれば 'A'} \end{cases}$

1111111111111111 を $\begin{cases} 2\text{の補数形式の数値として解釈すれば } -1 \\ \text{正の値のみの数値として解釈すれば } 65535 \end{cases}$

また、上で示したように文字データも 2 進数形式のデータとして扱われているため、「A」(0000000001000001) に 1 たすと「B」(0000000001000010) を示すというように、文字データに加算、減算等の演算を行うというようなことが可能である。

■例題8.1□ 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。(第34回)

[プログラムの説明]

二つのプログラム、MULT 01とMULT 02は、両方とも二つの自然数mとnを受け取り、その積m×nを求める副プログラムである。

- (1) mはGR 1に、nはGR 2に、それぞれ格納されて主プログラムから渡される。
- (2) 積m×nはGR 0に格納し、主プログラムに返す。
- (3) けたあふれなどは起きないものとする。つまり、積は10進数で32767を超えないものとする。

[プログラム01]

```

01 MULT 01 START
02 RPUSH
03 LAD GR 0, 0
04 LOOP SUBA GR 2, =1
05 JMI FIN
06 ADDA [①]
07 JUMP LOOP
08 FIN RPOP
09 RET
10 END

```

[プログラム02]

```

01 MULT 02 START
02 RPUSH
03 LAD GR 0, 0
04 LOOP LD GR 3, GR 2
05 [②] GR 3, =#0001
06 JZE OFF
07 ADDA GR 0, GR 1
08 OFF [③]
09 [④] GR 2, 1
10 JNZ LOOP
11 RPOP
12 RET
13 END

```

解説

プログラムの題意を一言で言えばGR 1 × GR 2 の答をGR 0 に代入する。これを実現する方法を知らない場合は、解答に手間取ることになるため、主な解法について、十分学習する必要がある。

GR 1が10(000000000001010), GR 2が5(000000000000101)としたときの積の求め方を示す。

考え方A

$$\begin{array}{r}
 & 1010 \\
 & 1010 \\
 & 1010 \\
 & 1010 \\
 +) & \hline
 & 1010 \\
 & \hline
 110010
 \end{array}
 \left. \right\} \text{GR1(000000000001010)を5(GR2)回足し算する。}$$

■考え方■：B

$$\begin{array}{r}
 1010 \quad \leftarrow \text{GR1} \\
 \times) \quad 101 \quad \leftarrow \text{GR2} \\
 \hline
 1010 \\
 0 \\
 \hline
 1010 \\
 \hline
 110010
 \end{array}
 \left.
 \begin{array}{l}
 \text{GR0(0000000000001010)を左に1ビットずらしながら加算する。} \\
 \text{ただし, GR2のビットを左からチェックしていき, 該当ビットが} \\
 \text{0のときは加算は行わない。}
 \end{array}
 \right\}$$

プログラム01の各命令を簡単に説明すると次のようになる。

01	MULT01	START	\leftarrow プログラムの開始
02		R PUSH	\leftarrow GR1～GR7の内容を退避保存
03		LAD	GR0, 0 \leftarrow レジスタGR0に0を設定
04	LOOP	SUBA	GR2, =1 \leftarrow レジスタGR2から1引く
05		JMI	FIN \leftarrow 直前のSUBA演算がマイナスならFINへジャンプ
06		ADDA	①
07		JUMP	LOOP \leftarrow 無条件にLOOPにジャンプ
08	FIN	RPOP	\leftarrow GR1～GR7の内容を元の値に復元
09		RET	\leftarrow 主プログラムに戻る
10		END	\leftarrow プログラム記述部の最後

GR2の値が、最初のGR2の値から-1まで変化し、GR2の初期値と同じ回数だけ繰り返すことから、考え方Aの足し算タイプと予測することができる。よって①に入るものは、最初にGR0が0で初期化されていることから、ADDA命令で、GR0にGR1を加算していることが予測できる。

次に、プログラム02の各命令を説明すると次のようになる。

01	MULT02	START	
02		R PUSH	
03		LAD	GR0, 0 \leftarrow レジスタGR0に0を設定
04	LOOP	LD	GR3, GR2 \leftarrow GR3にGR2の値を代入
05		②	GR3, =#0001
06		JZE	OFF \leftarrow 直前の演算結果が0と等しいならOFFへジャンプ
07		ADDA	GR0, GR1 \leftarrow GR0とGR1の加算結果をGR0に設定
08	OFF	③	GR1, 1
09		④	GR2, 1
10		JNZ	LOOP \leftarrow 直前の演算結果が0と等しくないならLOOPへジャンプ
11		RPOP	
12		RET	
13		END	

残っているプログラムがBの足し算タイプであることに気がつけば、GR1をさわっているのは、③だけなのでここにSLLが入ることは比較的簡単にわかる。次に、「GR2のビットを左からチェックしていき...」を実現する方法としては、GR2を右へ1ビットずつずらしながら、そのときどきで最右端のビットを#0001(0000000000000001)とANDすることで、1ビットを取り出しチェックする方法が一般的で、今回はSRLを②にANDを無理なく入れることができる。

このように、CASM II等のアセンブリ言語は基本的なプログラムのやり方（アルゴリズム）を知らなくては解けない問題も多い、本節では特に基本的な問題を中心に取り上げている。

解答

①GR0, GR1 ②AND ③SLL ④SRL

練習問題

1 CASL II プログラムについて答えなさい。

(1) 次のプログラムAの03行から06行を入れ替え、プログラムAと同じ実行結果が得られるプログラムBを作成した。プログラムB中の□を埋めてプログラムを完成させなさい。

(2) 間違って、GR1に3277(0000 1100 1100 1101)を設定してプログラムAを実行した。実行後GR1の値を2進数でいくらになるか答えなさい。

[プログラムAの説明]

(1) プログラムAは、GR1の値を10倍にしてGR1に設定する副プログラムである。

(2) GR1に設定される値は0~3276とする。

[プログラムA]

```

01 TMSA      START
02          PUSH   0, GR2
03          LD     GR2, GR1
04          SLA    GR1, 2
05          ADDA   GR1, GR2
06          SLA    GR1, 1
07          POP    GR2
08          RET
09          END

```

[プログラムB]

```

01 TMSB      START
02          PUSH   0, GR2
03          LD     ①
04          LD     ②
05          LD     ③
06          LD     ④
07          POP    GR2
08          RET
09          END

```

2 次のCASL II プログラム中の□を埋めてプログラムを完成させなさい。

[プログラムの説明]

GR1に格納された番地から始まり、GR2で示される語数からなる領域中の数値の最大値及び最小値を求める副プログラムである。

(1) 数値は、-32768~32767の範囲の値とし、それぞれ1語に格納されている。

(2) 結果は、GR2に最大値、GR3に最小値をセットし主プログラムに戻る。

(3) 領域の語数は、1語以上とする。

[プログラム]

```

01 MAXMIN  START
02          LD     GR5, GR2      ; 語数を待避
03          LD     GR0, 0, GR1
04          LD     GR2, GR0
05          LD     GR3, GR0
06          LAD    ①
07 S1       CPA    GR4, GR5      ; 終了か
08          JZE
09          LAD    ②
10          LD     GR0, 0, GR1
11          CPA    GR0, GR2
12          JMI    ③
13          JZE
14          LD     GR2, GR0      ; 最大値
15          JUMP
16 S2       CPA    GR0, GR3
17          JPL    ④
18          JZE
19          LD     GR3, GR0      ; 最小値
20 S3       LAD    GR4, 1, GR4
21          JUMP   S1
22 SRET    RET
23          END

```

- 3 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第35回)

[プログラムの説明]

このプログラムは、主プログラムから渡される二つの数値に、次の通りに対応する演算を施し、結果をG R 0 に格納して主プログラムに返す副プログラムである。

(1) 主プログラムから渡す二つの値は、それぞれG R 1, G R 2 に格納してある。

(2) 二つの数値に施すべき演算の種類は、表のようにコード化してG R 3 に格納してある。なお、格納しているコードには誤りはないものとする。

(GR3)	(GR1), (GR2)に対する演算
0	(GR1)+(GR2)
1	(GR1)-(GR2)
2	(GR2)-(GR1)
3	(GR1)と(GR2)との大きい方
4	(GR1)と(GR2)との小さい方

(3) この副プログラムから戻るとき、汎用レジスタ G R 1 ~ G R 3 は、呼び出されたときの内容のまます。

[プログラム]

```

0 1  SUBRTN  START
0 2          PUSH    0, GR 3
0 3          LD      GR 0, GR 1 ; 二つの数値の演算準備
0 4          LD      [①]      ; } 該当演算処理に分岐
0 5          JUMP   0, GR 3 ; }
0 6  LB 0    ADDA   GR 0, GR 2 ; 加算
0 7          JUMP   LB 6
0 8  LB 1    SUBA   GR 0, GR 2 ; 減算 1
0 9          JUMP   LB 6
1 0  LB 2    LD      [②]
1 1          SUBA   GR 0, GR 1 ; 減算 2
1 2          JUMP   LB 6
1 3  LB 3    CPA    GR 0, GR 2 ; 大きい値
1 4          [③]    LB 6
1 5          JUMP   LB 5
1 6  LB 4    CPA    GR 0, GR 2 ; 小さい値
1 7          [④]    LB 6
1 8  LB 5    LD      GR 0, GR 2
1 9  LB 6    POP    GR 3
2 0          RET
2 1  LTBL   DC     LB 0
2 2          DC     LB 1
2 3          DC     LB 2
2 4          DC     LB 3
2 5          DC     LB 4
2 6          END

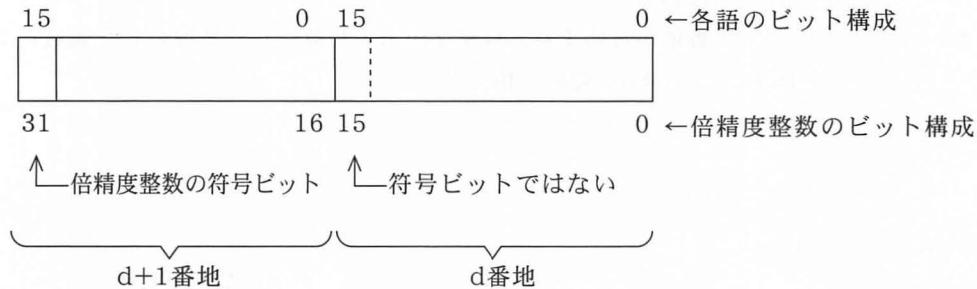
```

4 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第29回)

[プログラムの説明]

このプログラムは、倍精度整数の加算を行う副プログラムである。

(1) 倍精度整数Dは、図のように2語(32ビット)で構成されていて、d番地とd+1番地を占める。



(2) x番地とx+1番地を用いる倍精度整数X、y番地とy+1番地を用いる倍精度整数Y、z番地とz+1番地を用いる倍精度整数Zにおいて、倍精度整数の加算 $Z = X + Y$ は、次の手順で行う。なお、x番地の内容を(x)で示すものとする。

$$\textcircled{1} (x) + (y) = (z)$$

\textcircled{2} \textcircled{1} でけた上がりがないかどうか調べる。

$$\textcircled{3} \text{けた上がりの値} + (x+1) + (y+1) = (z+1)$$

(3) 主プログラムは、GR1にパラメタの先頭番地を格納して副プログラムを呼ぶ。パラメタの内容は以下のとおりとなる。ただし、(GR1)はGR1の内容を表している。

パラメタ	
	(GR1+0)
+1	x 番 地
+2	y 番 地
	z 番 地

[プログラム]

```

0 1 DBLADD     START
0 2               R P U S H
0 3               LD       G R 2, 0, G R 1 ; x のアドレスを G R 2 に格納
0 4               LD       G R 3, 1, G R 1 ; y のアドレスを G R 3 に格納
0 5               LD       [①] ; z のアドレスを G R 4 に格納
0 6               LD       G R 1, 0, G R 2 ;
0 7               ADDL     G R 1, 0, G R 3 ; } (x)+(y) → (z)
0 8               ST       [②] ;
0 9               JOV      C A R R Y
1 0               LAD      [③]
1 1               J U M P   N O C A R R Y
1 2   C A R R Y   LAD      G R 1, 1
1 3   N O C A R R Y   ADDA     [④] ;
1 4               ADDA     G R 1, 1, G R 3 ; } けた上げ+(x+1)+(y+1) → (z+1)
1 5               ST       G R 1, 1, G R 4 ;
1 6               R P O P
1 7               R E T
1 8               E N D

```

5 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第32回)

[プログラムの説明]

ある領域中の数値を調べる副プログラムである。

- (1) 負の数値の場合、絶対値に変換するとともに、その個数を数える。
- (2) 負の数値の個数を、 G R 0 に格納する。
- (3) 調べる数値は、 G R 1 が示す番地から始まり、 G R 2 が示す語数からなる連続した領域にある。
- (4) 格納している数値は-16384～16383の範囲の値である。

[プログラム]

0 1	S R C H	S T A R T	
0 2		R P U S H	
0 3		L A D	G R 0 , 0 ; 語数を 0 にセット
0 4		L D	G R 2 , G R 2
0 5	S 1	J Z E	① ; 終わりか
0 6		L D	G R 3 , ②
0 7		L D	G R 3 , G R 3
0 8		J P L	S 2
0 9		J Z E	S 2
1 0		A D D A	G R 0 , = 1 ; 語数に 1 を加える
1 1		X O R	G R 3 , = # ③ ; ビットを反転する
1 2		L A D	G R 3 , 1 , G R 3
1 3		S T	G R 3 , 0 , G R 1
1 4	S 2	L A D	④
1 5		S U B A	G R 2 , = 1
1 6		J U M P	S 1
1 7	S R E T	R P O P	
1 8		R E T	
1 9		E N D	

6 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

二つの語の内容を交換する副プログラム S W A P である。

- (1) 主プログラムは、二つの語のアドレスを G R 1 , G R 2 に設定して、 S W A P を呼ぶ。
- (2) S W A P は、 G R 1 , G R 2 でアドレスが指定された語の内容を交換する。
- (3) S W A P から戻るとき、汎用レジスタの内容は呼び出される前の値に復元する。

[プログラム]

0 1	S W A P	S T A R T	
0 2		P U S H	①
0 3		L D	②
0 4		P U S H	0 , G R 3
0 5		L D	G R 0 , 0 , G R 1
0 6		L D	③
0 7		S T	④
0 8		S T	G R 3 , 0 , G R 1
0 9		P O P	G R 0
1 0		P O P	G R 3
1 1		R E T	
1 2		E N D	

7 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第31回)

[プログラムの説明]

この副プログラムは、与えられた15ビットのビット列データに対して、パリティビットを求めるプログラムである。

- (1) 求めるパリティビットは偶数パリティで、与えられた15ビットにおいて1であるビットの個数が奇数ならばパリティビットは1となり、偶数(0を含む)ならばパリティビットは0となる。
- (2) 主プログラムはGR0の第0ビットから第14ビットにビット列データを設定して、この副プログラムに渡す。
- (3) 求めたパリティビットは、GR0の第15ビットに設定して主プログラムに戻される。
- (4) 渡されるGR0の第15ビットは0である。

例

GR0	0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0	→	0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0
GR0	0 1 1 0 0 1 1 1 1 0 0 1 1 1 0 0	→	1 1 1 0 0 1 1 1 1 0 0 1 1 1 0 0

[プログラム]

0 1	P A R I T Y	S T A R T	
0 2		R P U S H	
0 3		L A D	G R 3, 1 5 ; ループカウンタの設定
0 4		L A D	G R 2, 0 ; カウンタの初期設定
0 5		L D	[①] ; G R 0 の退避
0 6		L D	G R 1, G R 0
0 7	L 1	A N D	G R 0, = 1
0 8		A D D A	G R 2, G R 0
0 9		S U B A	[②]
1 0		J M I	L 2
1 1		S R L	[③]
1 2		L D	G R 0, G R 1
1 3		J U M P	L 1
1 4	L 2	L D	G R 0, G R 4 ; G R 0 の復元
1 5		A N D	G R 2, = 1
1 6		J Z E	L 3
1 7		X O R	[④]
1 8	L 3	R P O P	
1 9		R E T	
2 0		E N D	

8 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第28回)

[プログラムの説明]

1語を16ビットのビット列と考えて、指定されたビット群が1か0かをテストする副プログラムである。

(1) テストする1語は、G R 1に格納して主プログラムから渡される。

(2) テストすべきビット群は、G R 2(マスク)内の対応する位置のビットを1にして主プログラムから渡される。

(3) テスト結果は、G R 0に次の値を設定して主プログラムに戻す。

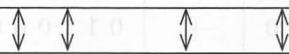
0 : 指定されたビットのすべてが0

1 : 指定されたビットの一部が1

2 : 指定されたビットのすべてが1

例 G R 1

1 0 0 0 0 0 1 1 0 0 1 0 1 1 0 0



G R 2(マスク)

0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0

G R 0(テスト結果)の内容 : 1

(4) G R 2が#0000であるときは、テスト結果を0とする。

[プログラム]

```

0 1 TESTBIT START
0 2      PUSH #0, G R 1
0 3      AND G R 1, G R 2
0 4      J Z E [①]; すべて0か?
0 5      X OR G R 1, G R 2
0 6      J Z E [②]; すべて1か?
0 7      L AD G R 0, 1 ; } 戻り値の設定
0 8 FIN    P OP G R 1 ; }
0 9      R ET ; }
1 0 P 1    L AD G R 0, 0 ; } 戻り値の設定
1 1      J U MP [③];
1 2 P 2    L AD G R 0, 2 ;
1 3      J U MP [④];
1 4      E ND

```

9 次のC A S L II プログラムの各間に答えなさい。

(1) 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

(2) 処理対象のデータ領域の内容が、16進表示で4000[0001]の2語のとき、L O O P 2の命令(J Z E)は何回実行されるか、その回数を答えなさい。

[プログラムの説明]

データ中の1ビットの個数を数える副プログラムである。データは、G R 1に格納された番地から始まり、G R 2で示される語数の領域にある。

(1) 求めた結果は、G R 3に格納する。

(2) 結果は、32767を越えない。

(3) G R 1, G R 2, G R 4の内容は保存されないものとする。

[プログラム]

```

0 1 BITCNT START BEGIN
0 2 SAVE DS 1
0 3 BEGIN ST G R 0, SAVE ; G R 0 の保存

```

0 4	L A D	G R 3 , 0	; G R 3 の初期化
0 5	L D	①	
0 6	L O O P 1	C P A	G R 2 , G R 4 ; 終了か?
0 7		J Z E	L A S T ; 終了なら最後へ
0 8		L D	G R 0 , 0 , G R 1
0 9	L O O P 2	J Z E	L B L 2 ; G R 0 が 0 なら 1 語分の処理終了
1 0		J P L	②
1 1		L A D	G R 3 , 1 , G R 3 ; ビットカウンタに 1 を加算
1 2	L B L 1	S L L	G R 0 , 1 ; G R 0 を左に 1 ビット論理シフト
1 3		J U M P	L O O P 2
1 4	L B L 2	L A D	G R 1 , 1 , G R 1 ; } 次の語の処理
1 5		L A D	③ ; } のための準備
1 6		J U M P	L O O P 1
1 7	L A S T	L D	G R 0 , S A V E ; G R 0 の復元
1 8		R E T	
1 9		E N D	

10 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

- (1) このプログラムは、キーボードからの入力文字列中の数字（0～9）の個数を数えて、表示するプログラムである。
- (2) 入力文字数は 10 文字未満とし、リターンキーだけが押されたときや、10 文字以上は再入力させる。ただし、その場合、エラーメッセージは出さない。

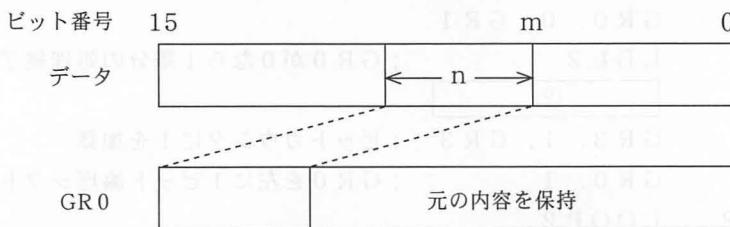
[プログラム]

0 1	I N N U M	S T A R T	
0 2	I N P U T	I N	B U F , L E N ; キーボードからの入力
0 3		L A D	G R 1 , 0
0 4		C P A	G R 1 , L E N
0 5		①	I N P U T ; リターンキーだけのとき再入力
0 6		L A D	G R 0 , 9
0 7		C P A	G R 0 , L E N
0 8		J M I	I N P U T ; 10 文字以上のとき再入力
0 9		L A D	G R 2 , 0
1 0	C H E C K	L D	G R 0 , B U F , G R 1 ; 入力文字列から 1 文字取り出す
1 1		C P L	G R 0 , = , '0'
1 2		②	N E X T ; コードが'0'より小さければカウントしない
1 3		C P L	G R 0 , = , '9'
1 4		③	N E X T ; コードが'9'より大きければカウントしない
1 5		L A D	G R 2 , 1 , G R 2
1 6	N E X T	L A D	④
1 7		C P A	G R 1 , L E N
1 8		J M I	C H E C K ; 入力文字を全てチェックしたか
1 9		⑤	G R 2 , = , '0' ; カウントした数を数字(文字)に変換
2 0		S T	G R 2 , C N T
2 1		O U T	C N T , M L E N ; 結果の出力
2 2		R E T	
2 3	B U F	D S	2 5 6 ; 入力文字が入る領域
2 4	L E N	D S	1 ; 入力文字数が入る領域
2 5	C N T	D S	1
2 6	M S G	D C	' M O J I D E S U '
2 7	M L E N	D C	9
2 8		E N D	

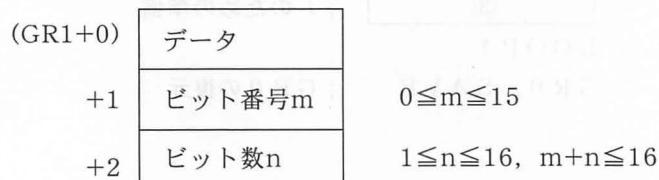
11 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第30回)

[プログラムの説明]

1語からなるデータのビット番号mから始まるnビットの値を取り出し、GR0の上位nビットにその値を設定する副プログラムFGSUBです。図にデータとGR0の関係を示す。



(1) GR1にパラメタの先頭アドレスを格納してFGSUBを呼ぶ。パラメタの形式は、図の通りとなる。



(2) FGSUBは、パラメタで指定されたデータのビット番号mから始まるnビットの値を取り出し、GR0にその上位 $16-n$ ビットの内容を保持したまま、左詰で格納する。

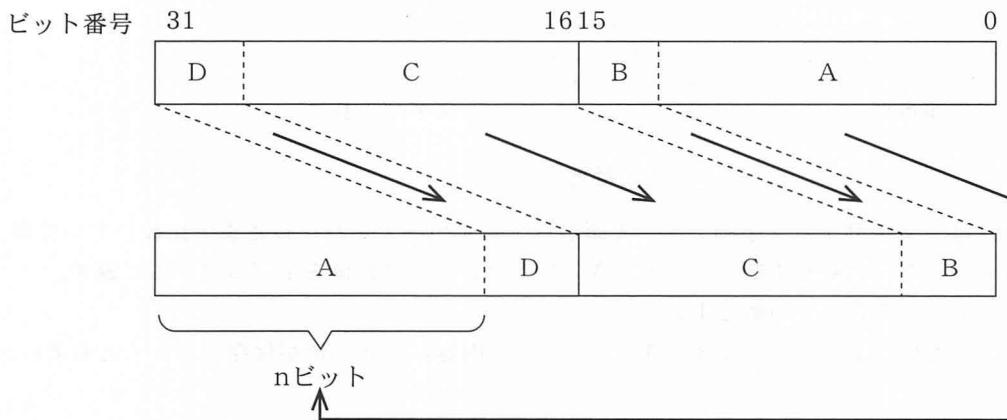
[プログラム]

```
01 FGSUB START
02      RPUSH
03      LD    [①]
04      LD    GR0, 0, GR1
05      LD    GR2, 1, GR1 ; m
06      SRL   GR0, 0, GR2
07      LAD   GR3, #FFFF
08      LD    GR2, 2, GR1 ; n
09      SRL   [②]
10      AND   GR3, GR4
11      LAD   GR2, 16
12      SUBA  [③] ; 16-n
13      SLL   GR0, 0, GR2
14      OR    [④]
15      RPOP
16      RET
17      END
```

12 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第33回)

[プログラムの説明]

連続する2語を32ビットのビット列と見なす。このビット列を指定されたビット数nだけ右に循環シフトする副プログラムS H I F Tである。右に循環シフトするとは、図に示すように、シフトの結果、右にはみ出したビット列を左端に格納することを言う。なお図において、ビット0~15のビット列が第1語、16~31ビット列が第2語になる。



- (1) ビット列の先頭番地はG R 1に、シフトするビット数はnはG R 2に格納され、主プログラムから渡される。
- (2) シフトした結果は、もとの領域に格納される。
- (3) $0 \leq n \leq 16$ とする。

[プログラム]

```

0 1   S H I F T      S T A R T
0 2               R P U S H
0 3       LD          G R 0, 0, G R 1
0 4       C A L L     D I V I D E      ; 第1語を二つの部分に分ける
0 5       LD          G R 4, G R 0
0 6       LD          G R 5, G R 3
0 7       LD          G R 0, 1, G R 1
0 8       C A L L     D I V I D E      ; 第2語を二つの部分に分ける
0 9       O R          ①
1 0       S T          G R 3, 0, G R 1
1 1       O R          ②
1 2       S T          G R 0, 1, G R 1
1 3       R P O P
1 4       R E T
1 5       E N D
1 6   ;
1 7   D I V I D E    S T A R T
1 8       S T          G R 2, S A V E
1 9       L D          ③
2 0       S R L         G R 0, 0, G R 2
2 1       L A D         G R 2, 16
2 2       S U B A       ④
2 3       S L L         G R 3, 0, G R 2
2 4       L D          G R 2, S A V E
2 5       R E T
2 6   S A V E        D S           1
2 7       E N D

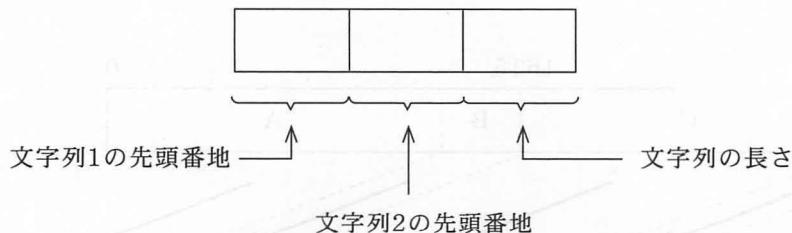
```

13 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

長さの等しい二つの文字列を比較して、結果に従ってF R(フラグレジスタ)に値をセットする副プログラムである。

- (1) 主プログラムにより二つの文字列の先頭番地及び長さが、次のような3語からなるテーブルに格納され、そのテーブルの先頭番地がG R 3にセットされて、この副プログラムに渡される。



- (2) この副プログラムは、文字列1と文字列2を、先頭から1語ずつ与えられた長さまで比較していくが、途中でもし等しくなくなったら比較を中断し、そのときのフラグレジスタの値を主プログラムに返す。
(3) 文字列の長さは、1～255までの値とする。
(4) この副プログラムの実行によって、G R 3以外のレジスタの内容は、元の値が保存されなくともよい。

[プログラム]

```
01 SHCOMP START
02 LD      GR4, 2, GR3 ; } 文字列1の最終番地 + 1 → GR4
03 ADDL   ①
04 LD      ②
05 LD      GR2, 1, GR3 ; 文字列2の先頭番地 → GR2
06 LOOP   LD      GR0, 0, GR1
07 CPL    GR0, 0, GR2 ; 文字列1の文字 : 文字列2の文字
08 JNZ    ③
09 LAD    GR1, 1, GR1 ; 文字列1の指標を1進める
10 LAD    GR2, 1, GR2 ; 文字列2の指標を1進める
11 CPA    GR1, GR4 ; 文字列の走査は終わったか？
12 JNZ    ④
13 LAST   RET
14 END
```

14 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第36回)

[プログラムの説明]

メモリ領域には、1語の正の整数(1～32767)を収容するための1次元配列が確保されており、それぞれテーブルA、テーブルB、テーブルCとする。このプログラムは、正の整数が格納されているテーブルA、Bから、両テーブルに共通する数値を抜き出して、テーブルCに格納する副プログラムである。

- (1) テーブルA、Bともに、テーブル内の正の数は昇順に格納されており、テーブル内での重複はない。
(2) テーブルA、Bの最後には、-1が格納されている。同様に、テーブルCの最後にも-1を格納する。
(3) 主プログラムは、G R 1にパラメタの先頭番地を格納して副プログラムを呼ぶ。パラメタの内容は図のとおりである。ただし、(G R 1)はG R 1の内容を表している。

パラメタの内容

(GR1)+0	テーブルAの先頭番地
+1	テーブルBの先頭番地
+2	テーブルCの先頭番地

[プログラム]

01	MATCH	START		
02		ST	GR 0, GOSAVE ; } レジスタの待避	
03		R P U S H	;	
04		L D	GR 2, 1, GR 1 ; テーブルBの先頭番地をGR 2に格納	
05		L D	GR 3, 2, GR 1 ; テーブルCの先頭番地をGR 3に格納	
06		L D	GR 1, 0, GR 1 ; テーブルAの先頭番地をGR 1に格納	
07	LOOP	L D	GR 0, 0, GR 2	
08		C P A	GR 0, = -1 ; Bの最後?	
09		J Z E	LAST	
10		L D	GR 0, 0, GR 1	
11		C P A	GR 0, = -1 ; Aの最後?	
12		J Z E	LAST	
13		C P A	GR 0, [①] ; Aの内容とBの内容の比較	
14		J P L	G T	
15		J Z E	E Q	
16	LT	L A D	GR 1, 1, GR 1 ; A < B	
17		J U M P	LOOP	
18	GT	L A D	GR 2, 1, GR 2 ; A > B	
19		J U M P	LOOP	
20	EQ	S T	[②], 0, GR 3 ; A = B	
21		L A D	GR 1, 1, GR 1	
22		L A D	GR 2, 1, GR 2	
23		L A D	[③], 1, GR 3	
24		J U M P	LOOP	
25	LAST	S T	GR 0, [④] ; Cの最後を格納	
26		R P O P		;
27		L D	GR 0, GOSAVE ; } レジスタの復元	
28		R E T		
29	GOSAVE	D S	1	
30		E N D		

〔コラム〕

CASL II プログラミングでは、主記憶上のアドレスにラベルと呼ばれる名前を付けることができる。このラベルと汎用レジスタ GR 1～GR 7 の内の 1 つとの組み合わせで、ラベル名のアドレスから汎用レジスタの値だけ下がった（増えた）アドレスを表現することが多い。命令でラベル名の次に GR 1～GR 7 の記述がある場合は、二つをくっつけて考えるとよい。

例 LD GR0, [TEST, GR4]
(GR4 の値が 2 なら)

ラベル名

TEST →

XXXXXXXXXXXX0000	A
XXXXXXXXXXXX0001	B
XXXXXXXXXXXX0010	C

このアドレスを示す。

8. 2 CASL II の応用的なプログラム

要 点

CASL II プログラムは 1 命令 1 動作なので、少し複雑なプログラムを作ると、長いプログラムとなってしまう。このような長文のプログラムの解析をする場合は、汎用レジスタやフラグレジスタの値の変化を 1 命令ごとにトレースすることが、案外と近道となることが多い。

例として、GR 1 には文字列データの先頭番地、GR 2 には文字列の長さを与え、文字列の長さが 1 ~ 20 の範囲でないときは、エラーとするプログラムの場合のトレースを示す。

```

01  CHKCHA   START
02          LAD    GR 3, 0
03          LD     GR 0, GR 2
04          JZE   ERROR
05          JMI   ERROR
06          CPA   GR 0, = 20
07          JPL   ERROR
08  LOOP 1   LD    GR 0, 0, GR 1

```

	GR 0	GR 1	GR 2	GR 3
		データの先頭番地	9	
02		↓	↓	0
03	9	↓	↓	↓
04	↓	↓	↓	↓
05	↓	↓	↓	↓
06	↓	↓	↓	↓
07	↓	↓	↓	↓
08	先頭のデータ	↓	↓	↓

□例題8.2□

次のCASL II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

- (1) このプログラムは、入力装置から 1 レコードの文字データを入力し、英大文字はそのままで、それ以外の文字は”？”に変換して表示する。
- (2) 例 入力データ # J Y 1 2 K
結果の表示 ? J Y ? ? K
- (3) 空のレコードが入力されるか又は EOF が検出されたとき、このプログラムは終了する。

[プログラム]

```

01  INCH      START
02  INPUT     IN      BUF, LEN      ; 1 レコードの入力
03          LD     GR 1, LEN
04          SUBA    [①]           ; 終了判定
05          JMI    FIN
06          LAD    GR 0, 63       ; '?' の文字コード
07          LAD    GR 2, 65       ; 'A' 文字のコード
08          LAD    GR 3, 91       ; 'Z' の 1 つ後の文字コード
09  LOOP      CPA    GR 2, BUF, GR 1
10          JPL    [②]
11          CPA    GR 3, BUF, GR 1
12          JPL    [③]
13  CHG       ST     GR 0, BUF, GR 1
14  NEXT     SUBA   GR 1, = 1
15          JPL    LOOP
16          JZE    [④]
17          OUT    BUF, LEN      ; 結果の表示
18          JUMP   INPUT

```

19	FIN	RET
20	BUF	DS
21	LEN	DS
22		END

256

1

解説

05行目のジャンプ命令がSF(Sign Flag)レジスタが負を示すときに、ラベルFINにジャンプすることと、題意の「空のレコードが入力されるか又はEOFが検出されたとき、このプログラムは終了する。」を合わせて考えると、データ領域LENに0又は-1が入力されたときにFINへジャンプするよう①を考えればよい、よってここにはGR1を1引く動作、GR1, =1を入れればよいことが解る。

少し難しいと思われる問題を分析するときは、要点にも述べたがトレースするのがよく、次に本プログラムに「#JY12K」が入力されたときのトレースを示す。

01	MAIN	START
02	INPUT	IN
03		LD
04		SUBA
05		JMI
06		LAD
07		LAD
08		LAD
09	LOOP	C PA
10		JPL
11		C PA
12		JPL
13	CHG	ST
14	NEXT	SUBA
15		JPL
16		JZE

			GR 0	GR 1	GR 2	GR 3
01	MAIN	START				
02	INPUT	IN				
03		LD				
04		SUBA	GR1, =1		6	
05		JMI	FIN		5	
06		LAD	GR0, 63		↓	
07		LAD	GR2, 65	'?',	↓	
08		LAD	GR3, 91	↓	↓	'A'
09	LOOP	C PA	GR2, BUF, GR1	↓	↓	'Z' +1
10		JPL	②	↓	↓	↓
11		C PA	GR3, BUF, GR1	↓	↓	↓
12		JPL	③	↓	↓	↓
13	CHG	ST	GR0, BUF, GR1	↓	↓	↓
14	NEXT	SUBA	GR1, =1	↓	↓	↓
15		JPL	LOOP	↓	4	↓
16		JZE	④	↓	↓	↓

これにより、②、③はジャンプ命令なのでGR1の値変化には関係しないことが解る。また、09行、11行、13行目のGR1の使用法から、ラベルBUF+GR1のアドレスにあるIN命令で入力した文字をアクセスしていることがプログラムから読める。よって、繰り返しのキーとなる14行目SUBA命令終了時点のGR1の値は4, 3, 2, 1, 0のときLOOPにジャンプし負の値のときに次の入力のためINPUTへジャンプすることが解る。よって④にLOOPが入ることが見えてくる。

②、③に付いてはC PA命令直後のジャンプ命令なのでBUF+GR1の文字データが大文字かそれ以外かの判別する部分であることが解る。図示すると次のようになる。



ここで注意することは、直前のC PA命令で第1オペランドのGR2('A')やGR3('Z'+1)と第2オペラント+第3オペラントのBUF+GR1の文字データを比較しているのだが、あくまでも第1オペラントのGR2, GR3が第2, 第3オペラントのBUF+GR1の文字データに比べ大きいか、等しいか、小さいかがSFフラグレジスタに設定される。これを逆にとらえると、当然ながら②にCHG, ③にNEXTを導き出すことが出来ない。

解答

①GR1, =1 ②CHG ③NEXT ④LOOP

練習問題

15 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第28回)

[プログラムの説明]

このプログラムは、8けたの数字データを入力し、ゼロ抑制処理を行った後、結果を表示するプログラムである。ゼロ抑制処理とは、先頭にある'0'を、'0'以外の数字が現れるまで間隔文字(スペース)に置き換える処理をいう。

(1) 例 入力データ 0 0 0 0 1 2 3 4

結果の表示 △△△△1 2 3 4 △は間隔文字を示す。

(2) 入力けた数が8けた以外のときは、エラーを表示し、再入力する。

(3) 全けた'0'のデータを入力したときは、最後の'0'だけを表示する。

(4) 復帰符号だけを入力すると、このプログラムは終了する。

(5) 数字以外のデータは入力はされないものとする。

[プログラム]

```

01  CONT      START   BEGIN
02  LENG      DS      1
03  INBUF     DS      2 5 6
04  ERLEN     DC      5
05  ERMSG     DC      ' ERROR'
06  BEGIN     IN      INBUF, LENG ; データの入力
07  LD        [①]
08  JZE       LAST
09  CPA       GR 3, = 8 ; 入力けた数=8 ?
10  JZE       GOOD
11  OUT      ERMSG, ERLEN ; 入力エラー
12  JUMP     [②]
13  GOOD     LD      GR 0, = ' 0' ; } レジスタの初期設定
14  LAD       [③]
15  LD        GR 2, = ' '
16  LOOP     CPA    GR 1, = 7
17  JZE       OUTPUT
18  CPL      [④] ; 入力データ='0' ?
19  JNZ       OUTPUT
20  ST        [⑤] ; ゼロ抑制
21  LAD      GR 1, 1, GR 1 ; 入力データの指標更新
22  JUMP     LOOP
23  OUTPUT   OUT    INBUF, LENG ; 結果の表示
24  JUMP     BEGIN
25  LAST     RET
26  END

```



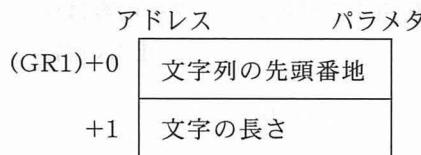
入力データ (DATA) の最初の4字と最後の1字を「最初の4字と最後の1字」の順で記述する。
ゼロ抑制 (ZERO-SUPPRESSION) は、最初の4字と最後の1字を「最初の4字と最後の1字」の順で記述する。
結果 (RESULT) は、最初の4字と最後の1字を「最初の4字と最後の1字」の順で記述する。

16 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第29回)

[プログラムの説明]

このプログラムは、文字の先頭、末尾および文字列中の単語間の二つ以上連続する空白 (△) を一つに詰める副プログラムである。

(1) 主プログラムは、G R 1 にパラメタの先頭番地を格納して副プログラムを呼ぶ。パラメタの内容は、図のとおりであり。(G R 1) は G R 1 の内容を表すものとする。



(2) 副プログラムは、文字列操作後、G R 1 を呼び出される前の内容に戻し、新しい文字列の長さを(G R 1) + 1 番地に格納して、主プログラムに戻る。

例 副プログラム呼び出し前

文字列 : △△△What△is△△△△△that?△△△△ 長さ = 24

副プログラム呼び出し後

文字列 : △What△is△that?△ 長さ = 15 (△は空白を示す)

[プログラム]

0 1	S P S P R S	STAR T	
0 2		P U S H	0, G R 1
0 3		L D	G R 2, 0, G R 1 ; G R 2 : 出力用アドレス
0 4		L D	G R 4, G R 2
0 5		A D D L	① ; 文字列の最後のアドレス + 1 → G R 4
0 6		L D	G R 1, 0, G R 1 ; G R 1 : 入力用アドレス
0 7		L A D	G R 3, 0 ; 空白検出フラグ 0
0 8	LO O P	L D	G R 0, 0, G R 1
0 9		C P L	G R 0, =, ' ; 空白か ?
1 0		J N Z	N O T S P C
1 1		L D	②
1 2		J N Z	N E X T
1 3		L A D	G R 3, 1
1 4		J U M P	M O V C H R
1 5	N O T S P C	L A D	③ ; 空白検出用フラグ 0
1 6	M O V C H R	S T	G R 0, 0, G R 2
1 7		L A D	G R 2, 1, G R 2
1 8	N E X T	L A D	G R 1, 1, G R 1
1 9		C P L	G R 1, G R 4
2 0		④	LO O P
2 1		P O P	G R 1
2 2		S U B L	⑤
2 3		S T	G R 2, 1, G R 1 ; 文字列の長さを格納
2 4		R E T	
2 5		E N D	

17 次のC A S L IIプログラム中の [] を埋めてプログラムを完成させなさい。 (第33回)

[プログラムの説明]

このプログラムは、主プログラムから渡された1語の内容を、ゼロ抑制を行った2進表記で出力する副プログラムである。

- (1) 出力する1語は、GR 1に格納して主プログラムから渡される。
- (2) 出力する際、先行する0のビットは出力しない。ただし、GR 1の全ビットが0のときは"0"を出力する。
- (3) GR 0を除いて、すべての副プログラムで使用する汎用レジスタの内容は保存される。

例

GR 1

出力の内容

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
0 0 0 0 1 0 1 0 1 1 1 1 0 0 0 0	1 0 1 0 1 1 1 1 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

[プログラム]

```

0 1 BITOUT START
0 2 RPUSH
0 3 LAD GR 2, 0 ; 出力ポインタの初期化
0 4 LAD GR 3, 15 ; カウンタの初期化
0 5 LD [①] ; 全ビット = 0 ?
0 6 JZE OUT 1
0 7 LOOP 1 JMI ONE ; } ; 先行するビット 0 の処理
0 8 LAD GR 3, -1, GR 3 ;
0 9 SLL [②] ;
1 0 JUMP LOOP 1
1 1 LOOP 2 JMI ONE ; 符号ビット判定
1 2 ZERO LAD GR 0, 48 ; '0' → GR 0
1 3 JUMP STORE
1 4 ONE LAD GR 0, 49 ; '1' → GR 0
1 5 STORE ST [③]
1 6 LAD GR 2, 1, GR 2
1 7 LAD GR 3, -1, GR 3
1 8 LD GR 3, GR 3
1 9 [④] OUT 2 ; 全ビット終了?
2 0 SLL GR 1, 1
2 1 JUMP LOOP 2
2 2 OUT 1 LAD GR 0, 48
2 3 ST GR 0, OBUF, GR 2
2 4 LAD GR 2, 1, GR 2
2 5 OUT 2 ST [⑤]
2 6 OUT OBUF, OLNG
2 7 RPOP
2 8 RET
2 9 OBUF DS 256
3 0 OLNG DS 1
3 1 END

```

18 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第35回)

[プログラムの説明]

主プログラムから与えられた10進数の文字列を、編集して出力する副プログラムEDTOUTである。

(1) 主プログラムは、GR1にパラメタの先頭番地を格納して副プログラムにEDTOUTを呼ぶ。パラメタの内容と意味は、次のとおりとする。(GR1)はGR1の内容を表している。

アドレス (GR1)+0	パラメタ 10進数の文字列の先頭アドレス
+1	10進数の文字列の長さ

(2) 文字列は"0"～"9"の文字からなる。

(3) 文字列中の各文字は、連続する語の下位8ビットに格納され、上位8ビットにはすべて0が入っている。

(4) E D T O U T は文字列に対して次の編集を行った後、その内容を出力する。

- 3 けたごとに、位取り(,)を挿入する。
 - 18語の出力領域に右詰めにする。
 - 位取りを挿入した文字列が18語に満たない場合は、空き領域に間隔文字を詰める。
 - 位取りを挿入した文字列が18語を超える場合は、右から18語の部分だけを出力する。

例 編集前の文字列

出力する文字列 (\triangle : 間隔文字を表す)

1 2 3 4

編集前の文字列

出力する文字列

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6

3 4, 5 6 7, 8 9 0, 1 2 3, 4 5 6

〔プログラム〕

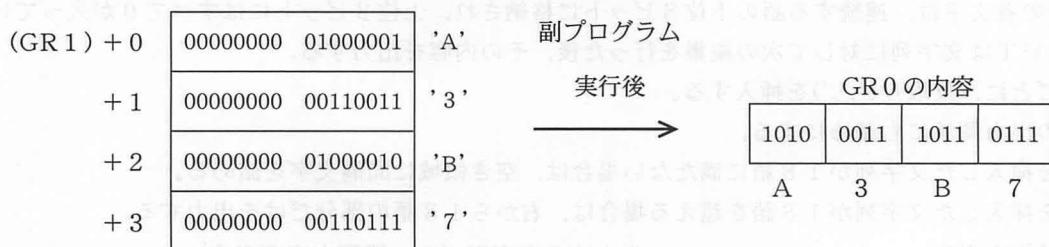
0 1	EDTOUT	START		
0 2		LD	G R 4, 0, G R 1	; 文字列の先頭アドレスを G R 4 へ
0 3		LD	G R 1, 1, G R 1	; 文字列の長さを G R 1 へ
0 4		ADDL	G R 1, G R 4	
0 5		LAD	G R 1, -1, G R 1	; 文字列の最後のアドレス
0 6		LD	G R 2, OUTLNG	; 出力領域の添字
0 7		LAD	G R 3, 3	; 位取りカウンタの初期化
0 8	LOOP	SUBA	G R 2, =1	; 出力領域の最後か？
0 9		JMI	(1)	
1 0		CPL	G R 1, G R 4	
1 1		JMI	SPCPAD	
1 2		SUBA	G R 3, =1	; 位取りの判断
1 3		JMI	(2)	
1 4		LD	G R 0, 0, G R 1	
1 5		LAD	G R 1, -1, G R 1	
1 6		JUMP	MOVCHR	
1 7	RANK	LD	G R 0, =', '	
1 8		LAD	(3)	; 位取りカウンタの再初期化
1 9		JUMP	MOVCHR	
2 0	SPCPAD	LD	G R 0, '=', '	
2 1	MOVCHR	ST	G R 0, OUTBUF, (4)	
2 2		JUMP	LOOP	
2 3	OUTPUT	OUT	OUTBUF, OUTLNG	
2 4		RET		
2 5	OUTBUF	DS	1 8	
2 6	OUTLNG	DC	(5)	
2 7		END		

19 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第31回)

[プログラムの説明]

文字データで与えられた4けたの16進数を、1語の2進数に変換する副プログラム C H 2 B I N である。

- (1) 各文字は、"0"~"9", "A"~"F"のいずれかとする。
- (2) 文字データは、連続する4語の領域に高位のけたから順に入れられ、その先頭アドレスがG R 1に格納されて主プログラムから渡される。各文字は各語の下位8ビットに格納され、上位ビットにはすべて0が入っている。
- (3) 第1語の文字から順に、それに対応する4ビットの2進数に変換し、G R 0に上位ビットから順に格納して、主プログラムに返す。
- (4) "A 3 B 7"の文字列が与えられた場合の変換例を次に示す。



[プログラム]

```

0 1  C H 2 B I N  S T A R T
0 2          R P U S H
0 3          L A D  S T  [①]
0 4          L A D  G R 2 , 4 ; ループカウント
0 5  L O O P  L D   G R 3 , 0 , G R 1
0 6          C P A  G R 3 , = ' A ' ; "0"~"9"?
0 7          J M I  [②]
0 8          A D D A G R 3 , = 9 ; } 文字を2進数に変換
0 9  N U M  [③] G R 3 , = # 0 0 0 F ;
1 0          S L L  G R 0 , 4 ; 変換済みの値をシフト
1 1          [④] G R 0 , G R 3 ; 現在の値とマージ
1 2          L A D  G R 1 , 1 , G R 1
1 3          L A D  G R 2 , - 1 , G R 2
1 4          L D   G R 2 , G R 2 ; 全文終了?
1 5          J N Z  [⑤]
1 6          R P O P
1 7          R E T
1 8          E N D

```

20 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第32回)

[プログラムの説明]

- (1) 試験の採点を行う副プログラム S A I T E N である。
- (2) 試験問題は16問あり、すべて解答すべきものである。
- (3) 解答は二者択一方式 (○×式) である。
- (4) 正解は定数として、S E I K A I で示される1語16ビットにあらかじめ格納されている。各問題の正解は、○は1、×は0として、1ビットで表す。
- (5) 解答も1語16ビットで表し、G R 1に格納して主プログラムから渡される。
- (6) 得点 (1問1点とする) は、G R 2に格納して、主プログラムに返す。

(7) 正解、解答とも、問題番号とビット位置は対応している。

例	問題番号	: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
	ビット位置	: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	正 解	: ○ × × × ○ × × × ○ × × × ○ × × ×
		S E I K A I : 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
	解 答	: ○ × ○ × ○ × ○ × ○ × ○ × ○ × ○ ×
	(G R 1)	: 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
	正 誤	: 正 正 誤 正 正 誤 正 正 正 誤 正 正 正 誤 正 (= 12 点)

[プログラム]

0 1	SAITEN	START	
0 2		PUSH	0, G R 1 ; レジスタの退避
0 3		PUSH	0, G R 3
0 4		LAD	① ; 得点を初期化
0 5		LAD	G R 3, 1 6 ; カウンタの初期値設定
0 6		②	G R 1, S E I K A I ; 解答と正解の比較
0 7	CHECK	LD	③ ; 探点
0 8		J M I	S K I P
0 9		LAD	④ ; 得点の加算
1 0	SKIP	⑤	G R 1, 1
1 1		S U B A	G R 3, = 1 ; カウンタの更新
1 2		J P L	C H E C K
1 3		P O P	G R 3 ; レジスタの復元
1 4		P O P	G R 1
1 5		R E T	
1 6	S E I K A I	D C	# 8 8 8 8 ; 正解
1 7		E N D	

〔コラム〕

長文のC A S L II プログラミングを解析する場合、プログラムのどの行からどの行までが一つの意味を持つブロックかを読みとることが大切である。C A S L II にはJ P L, J M I, J N Z, J Z E, J O V, J U M P の6つの分岐命令があるが、プログラムを読む前に、分岐命令の下に線を入れ、分けてから読むとブロック分けが楽である。

例として、G R 1には文字列データの先頭番地、G R 2には文字列の長さを2進数で与え、文字列の長さが1～20の範囲でかつ、文字が'間隔'～'Z'の59種類の文字で構成されていない場合はエラー処理をする場合を示す。

0 1	CHKCHA	START	
0 2		LAD	G R 3, 0
0 3		LD	G R 0, G R 2
0 4		J Z E	ERROR
0 5		J M I	ERROR
0 6		C P A	G R 0, = 2 0
0 7		J P L	ERROR
0 8	LOOP 1	LD	G R 0, 0, G R 1
0 9		C P L	G R 0, = # 0 0 2 0
1 0		J M I	ERROR
1 1		C P L	G R 0, = # 0 0 5 A

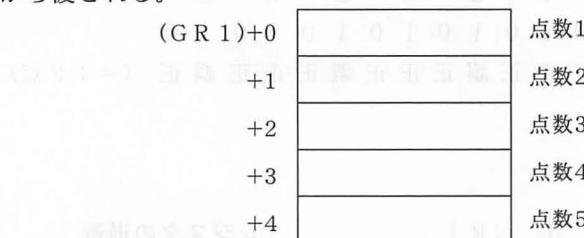
} 文字列の長さが0以下か？
} 文字列の長さが21以上か？
} 文字列データが#0020（間隔文字の一つ前）以下か？

21 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

5人の採点者のつけた点数を受け取り、最高点1つと最低点1つを除いた3つの点数の合計をG R 0に設定し呼び出しプログラムに戻す、副プログラムG R A Dである。

- (1) 5つの点数は、図のように連続する5つの語に入れられ、その先頭アドレスがG R 1に格納されて呼び出しへプログラムから渡される。



点数1

点数2

点数3

点数4

点数5

- (2) 点数は0以上、100点以下の整数とする。

- (3) 呼び出しプログラムに戻るとき、汎用レジスタG R 1～G R 6の内容を元に戻す。

[プログラム]

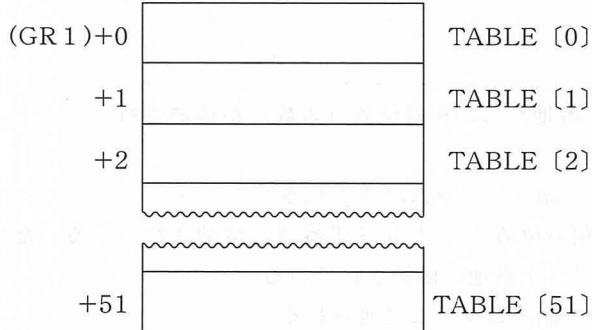
0 1	G R A D	START	
0 2		R P U S H	
0 3		L A D	G R 2, 1
0 4		L D	G R 0, 0, G R 1 ; 最初の点数をセット
0 5		L D	G R 3, G R 0
0 6		L D	G R 4, G R 0
0 7		L D	G R 5, G R 0
0 8		L A D	G R 6, 4 ; ループカウンタ
0 9	L O O P	L A D	G R 1, 1, G R 1
1 0		L D	G R 0, 0, G R 1
1 1		C P A	G R 0, G R 3
1 2		J P L	[①]
1 3		L D	G R 3, G R 0
1 4		J U M P	NOTMAX
1 5	N O T M I N	C P A	G R 0, G R 4
1 6		J M I	[②]
1 7		L D	G R 4, G R 0
1 8	N O T M A X	A D D A	G R 5, G R 0
1 9		S U B A	G R 6, G R 2
2 0		J N Z	[③]
2 1		L D	G R 0, G R 5
2 2		[④]	G R 0, G R 3
2 3		[⑤]	G R 0, G R 4
2 4		R P O P	
2 5		R E T	
2 6		E N D	

22 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第36回)

[プログラムの説明]

このプログラムは、52枚のトランプをシャッフルする(1～52の数をランダムに並べて、指定された配列領域にセットする)副プログラムである。

- (1) 主プログラムは、結果を格納すべき配列領域の先頭番地をG R 1にセットして、副プログラムS H U F Fを呼ぶ。



- (2) 副プログラムS H U F Fにおける並べ替えは、次に示す手順で行う。

a) 配列に1～52の値をこの順で初期設定する(T A B L E [0] = 1, T A B L E [1] = 2, T A B L E [2] = 3, ···)。

b) iを51から順に1まで変化させながら、次の動作を繰り返す。

T A B L E [i] とT A B L E [r] を交換する。ただし、rは $0 \leq r \leq i$ の整数乱数値とする。

- (3) 副プログラムS H U F Fは、内部で別の副プログラムR A N Dを使用する。R A N Dは、 $0 \sim (N - 1)$ の範囲の整数乱数を返す副プログラムである。G R 4にNを設定した後R A N Dを呼び出すと、R A N DはG R 4に整数乱数値を返す。R A N Dの実行によって、汎用レジスタの内容は変更されない。

- (4) 副プログラムS H U F Fから戻るとき、汎用レジスタの内容は、元に復元されなくてよい。

[プログラム]

```
01 SHUFF    START
02          LAD      GR 2, [①] ; 配列最終要素アドレス→G R 2
03          LAD      GR 3, 5 2
04 INITLP   [②]    GR 3, 0, GR 2 ;
05          LAD      GR 2, -1, GR 2 ;
06          LAD      GR 3, -1, GR 3 ; } 配列初期値設定
07          LD       GR 3, GR 3 ;
08          JNZ     INITLP ;
09          LAD      GR 2, 5 1, GR 1
10          LAD      GR 3, 5 1 ; 5 1 → i
11 LOOP     LAD      GR 4, [③] ;
12          CALL    RAND      ; 亂数(r)を生成,  $0 \leq r \leq i$ 
13          ADDL    GR 4, GR 1 ; T A B L E [ r ] のアドレス
14          LD       GR 0, 0, GR 4 ;
15          LD       GR 5, 0, GR 2 ;
16          ST       GR 5, [④] ;
17          ST       GR 0, [⑤] ; } T A B L E [ i ] とT A B L E [ r ] の交換
18          LAD      GR 2, -1, GR 2
19          LAD      GR 3, -1, GR 3 ; i - 1 → i
20          LD       GR 3, GR 3 ; i ≥ 1 ?
21          JNZ     LOOP
22          RET
23          END
```

23 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

指定されたデータと同じ値をもつ要素をリストから削除する副プログラムである。

(1) リストは、L I S T 番地から始まる連続した200語にあらかじめ格納されている。

(2) リストの各要素は、1語で構成され次のようにになっている。

ビット15

8 7

0

データ部	ポインタ部
------	-------

(3) ポインタ部は、次の要素のL I S T 番地からの相対位置（語数）が格納され、リストの最後の要素のポインタ部には、(F F)₁₆が格納されている。

(4) L I S T 番地は、G R 2 に格納されて副プログラムに渡される。

(5) リストの先頭の要素へのポインタ（相対位置）は、F R S T 番地に格納されている。ただし、リストの要素が存在しない場合は(0 0 F F)₁₆がF R S T 番地に格納されている。

(6) F R S T 番地は、G R 1 に格納されて副プログラムに渡される。

(7) 削除する要素のデータ部の値は、G R 3 のビット0～7に格納されて副プログラムに渡される。

(8) リストの中にはデータ部が同一の値をもつ要素が、存在することはない。

(9) 同一の値をもつ要素が存在した場合はG R 0 に0、存在しなかった場合はG R 0 に-1を格納して、主プログラムに戻る。

[プログラム]

```
01  DELE      START
02  RPUSH
03  SLL       G R 3, 8
04  ST         ①
05  LD         G R 3, 0, G R 1 ; 最初の要素へのポインタを得る
06  D10       CPA     G R 3, MASK 2 ; 最後の要素か
07  JZE       ②
08  ADDL     G R 3, G R 2
09  LD         G R 0, 0, G R 3
10  AND      G R 0, MASK 1 ; データを取り出す
11  CPL      G R 0, DATA
12  JZE       ③
13  LAD      G R 1, 0, G R 3 ; 一つ前の要素が格納されている番地を設定する
14  LD         G R 3, 0, G R 3
15  AND      G R 3, MASK 2 ; ポインタを取り出す
16  JUMP     ④
17  D20       LD      G R 4, 0, G R 3
18  AND      G R 4, MASK 2
19  LD         G R 0, 0, G R 1 ; } 一つ前の要素からデータだけを取り出す
20  AND      G R 0, MASK 1 ;
21  OR        G R 0, G R 4 ; 一つ前の要素にポインタを設定する
22  ST         G R 0, 0, G R 1
23  LAD      G R 0, 0
24  JUMP     ⑤
25  D30       LAD     G R 0, -1
26  D40       RPOP
27  RET
28  DATA      DS      1
29  MASK1    DC      # F F 0 0
30  MASK2    DC      # 0 0 F F
31  END
```

24 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

長さ 1 以上 4 以下の数字の列を受け取り、その数値を 2 進数に変換する副プログラム C O N V R T である。

- (1) 数字の列は、連続する n 語の領域に入れられ、その先頭アドレスが、 G R 1 に格納されて主プログラムから渡される。

各数字は各語の下位 8 ビットに格納され、上位 8 ビットにはすべて 0 が入っている。

- (2) 数字の列の直後の語 (n+1 語目) は、すべてのビットに 0 が入っている。

- (3) 数字の列を 10 進数とみなし、それを 2 進数に変換して G R 0 に格納し、主プログラムに返す。

例えば、数字の列' 1 2 3 4 ' が与えられた場合の変換は図のようになる。図では、領域の内容及び G R 0 の内容は 16 進数で表示してある。

変換の例



- (4) 次の場合、 G R 0 に -1 を格納して、主プログラムに返す。

- a) 数字以外の文字が現れた。
b) 数字の列の長さが 0 であった。
c) 数字の列の長さが 4 を超えた。

- (5) このプログラムから戻るとき、汎用レジスタ G R 1 ~ G R 7 を、呼び出される前の内容に復元する。

[プログラム]

0 1	C O N V R T	S T A R T		
0 2		R P U S H		
0 3		L D	G R 2, 0, G R 1	
0 4		J Z E	E R R O R	; 文字の長さ 0 ?
0 5		L A D	G R 3, 4	; 文字列長チェック用カウンタ
0 6		L A D	G R 0, 0	
0 7	L O O P	C P L	G R 2, Z E R O	; $\geq' 0' ?$
0 8		J M I	E R R O R	
0 9		C P L	G R 2, O V E R	; $\leq' 9' ?$
1 0		①	E R R O R	
1 1		A N D	G R 2, M A S K	
1 2		S L L	②	
1 3		S T	G R 0, W O R K	
1 4		S L L	③	
1 5		A D D A	G R 0, W O R K	
1 6		A D D A	G R 0, G R 2	
1 7		L A D	④	
1 8		L D	G R 2, 0, G R 1	
1 9		⑤	F I N	; 文字列の終わり ?
2 0		L A D	G R 3, -1, G R 3	
2 1		L D	G R 3, G R 3	
2 2		J N Z	L O O P	
2 3	E R R O R	L A D	G R 0, -1	; エラーコードの設定
2 4	F I N	R P O P		
2 5		R E T		
2 6	Z E R O	D C	# 0 0 3 0	; ' 0 '
2 7	O V E R	D C	# 0 0 3 9	; ' 9 '
2 8	M A S K	D C	# 0 0 0 F	
2 9	W O R K	D S	1	
3 0		E N D		

25 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

次の副プログラムは除算を行うものです。

- (1) 被除数がG R 1に、除数がG R 2に格納されてこの副プログラムを呼ぶ。商をG R 3、剩余をG R 0に格納して主プログラムに戻る。
- (2) 被除数の値が負または除数の値がゼロ以下である場合は、商に-1を格納する。
- (3) 除算は、次に示す手順で行う。
 - a) 作業領域を初期化する。
 - b) 最初の除数を格納する。
 - c) 商に1を格納する。
 - d) 商及び除数を1ビットずつ左にシフトする。
 - e) 除数が被除数より大きくなった場合、シフトをやめる。
 - f) 除数を1ビット右にシフトし、被除数から引き、結果を新しい被除数とする。
 - g) 商を1ビット右にシフトし作業領域に加える。
 - h) b～gの処理を、被除数が最初の除数より小さくなるまで繰り返す。

[プログラム]

```

0 1   D I V      S T A R T
0 2           P U S H    0, G R 1
0 3           S T       G R 2, W K      ; 除数の保存
0 4           L A D     G R 3, -1
0 5           L D       G R 1, G R 1
0 6           J M I     D R E T    ; 被除数は負か
0 7           C P A     G R 2, =1
0 8           J M I     [①]      ; 最初の除数はゼロ以下か
0 9           L A D     G R 3, 0
1 0           S T      G R 3, S H O    ; 商の初期化
1 1   D 1 0      L D      G R 2, W K
1 2           L A D     G R 3, 1
1 3   D 2 0      C P L     G R 1, G R 2
1 4           J M I     D 3 0
1 5           S L L     G R 3, 1      ; 商のシフト
1 6           [②]      G R 2, 1      ; 除数のシフト
1 7           J U M P   D 2 0
1 8   D 3 0      [③]      G R 3, 1
1 9           J Z E     D 4 0      ; 除算終了か
2 0           A D D A   G R 3, S H O
2 1           S T      [④]
2 2           S R L     G R 2, 1
2 3           S U B A   [⑤]      ; 新しい被除数のセット
2 4           J U M P   D 1 0
2 5   D 4 0      L D      G R 3, S H O    ; 商のセット
2 6           L A D     G R 0, 0, G R 1    ; 剩余のセット
2 7           L D      G R 2, W K
2 8   D R E T    P O P     G R 1
2 9           R E T
3 0   W K       D S      1
3 1   S H O     D S      1
3 2           E N D

```

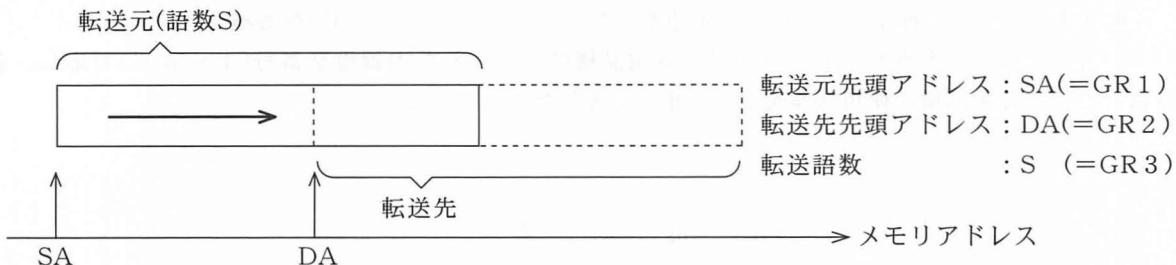
26 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。

[プログラムの説明]

次の副プログラムは、メモリーブロックを転送するための副プログラムである。

- (1) 主プログラムは、転送元ブロックの先頭アドレスをG R 1に、転送先ブロックの主アドレスをG R 2に、転送するブロックの語数をG R 3に設定してこの副プログラムを呼ぶ。

これらのアドレスと語数を、それぞれS A、D A、Sと略記する。



- (2) このプログラムから戻るとき、汎用レジスタG R 1～G R 7の内容は、元の値に復元する。

[プログラム]

```
0 1  MMOVE      S T A R T
0 2          R P U S H
0 3          L D      G R 3, G R 3
0 4          J Z E    F I N
0 5          C P L    G R 2, G R 1
0 6          J Z E    F I N
0 7          J M I    ①
0 8          L D      G R 4, G R 1      ; ブロック先頭から転送開始
0 9          A D D L   G R 4, G R 3
1 0          C P L    G R 4, G R 2
1 1          J P L    B R T N
1 2  A R T N    L D      G R 4, G R 1      ; ブロック先頭から転送開始
1 3          L D      G R 5, G R 2
1 4  A R T N L  L D      G R 0, 0, G R 4
1 5          S T      G R 0, 0, G R 5
1 6          L A D    ②
1 7          L A D    G R 5, 1, G R 5
1 8          L A D    ③
1 9          L D      G R 3, G R 3
2 0          J N Z    A R T N L      ; 転送終了?
2 1          J U M P   F I N
2 2  B R T N    L D      ④      ; ブロック最後尾から転送開始
2 3          A D D L   G R 5, G R 3
2 4  B R T N L  L A D    G R 4, -1, G R 4
2 5          L A D    ⑤
2 6          L D      G R 0, 0, G R 4
2 7          S T      G R 0, 0, G R 5
2 8          L A D    G R 3, -1, G R 3
2 9          L D      G R 3, G R 3
3 0          J N Z    B R T N L      ; 転送終了?
3 1  F I N     R P O P
3 2          R E T
3 3          E N D
```

27 次のC A S L II プログラム中の [] を埋めてプログラムを完成させなさい。 (第34回)

[プログラムの説明]

構内用交換機に接続されている電話機の使用状態を管理する副プログラムである。

- (1) 接続されている電話機は、全部で50台である。
- (2) 交換機に接続されている電話機同士の通話を内線通話と呼ぶ。交換機から外へ出るまたは外から入ってくる電話を外線通話と呼ぶ。
- (3) 各電話機の状態を管理するために、51語からなるテーブル(TABLE)がある。
- (4) (TABLE + 1)番地からの50語は、各電話機につけられた内線電話番号(1~50)に対応し、各語の値は、対応する電話機の使用状態をコード化して表したものである。

コード	使　用　状　態
0	未使用 (あき状態)
1~50	内線通話中 (コードの値は相手の電話番号)
100	外線通話中

- (5) 主プログラムは、内線電話番号をGR1に格納し、処理区分(通話開始要求、通話終了、通話中チェック)をGR2に格納して、この副プログラムを呼ぶ。

(GR 2)	処　理　区　分
0	通話終了
1~50	内線通話開始要求((GR 2)の値は相手電話番号)
100	外線通話開始要求
-1	通話中チェック

- (6) この副プログラムは、処理区分に応じて次の処理を行う。

- (GR 2) ≥ 0 のとき：テーブルの該当位置を更新する。なお、内線の場合は、相手電話のテーブルも更新する。
- (GR 2) = -1 のとき：通話中であれば -1 を、そうでなければ 0 を GR3 に格納する。

(GR 3)	結　果
0	未使用 (あき状態)
-1	通話中

[プログラム]

```

01 TELCHK START
02           LAD    GR 4, 51
03           ①     GR 2, GR 2
04           JPL    ②      ; 開始／終了？
05           JZE    SETSTS
06           LD     GR 3, TABLE, GR 1 ; 電話中チェック
07           JZE    ③
08           LAD    GR 3, -1      ; 通話中
09           JUMP   FIN
10 SETSTS  LD     GR 3, TABLE, GR 1
11           CPA    GR 3, GR 4      ; 内線／外線？
12           JPL    ④
13           ST     GR 2, TABLE, GR 3
14 CLS      ST     GR 2, TABLE, GR 1
15           JUMP   FIN
16 BGN      CPA    GR 2, GR 4
17           JPL    ⑤
18           ST     GR 1, TABLE, GR 2
19 EXTBGN  ST     GR 2, TABLE, GR 1
20 FIN      RET
21 TABLE   DS     1
22           DS     50
23           END

```

9. Cプログラムの作成能力

9. 1 Cの応用的なプログラム

要 点

[1] 関数

あるひと固まりの処理を行う処理単位である。関数には、あらかじめ処理系で用意されている標準ライブラリ関数と、プログラマが作成するユーザ関数がある。

[書式]

```
戻り値の型 関数名(仮引数宣言, ……)  
{  
:  
    return 戻り値;  
}
```

[例]

```
int d_max(int d1, int d2)  
{  
:  
    return kekka;  
}
```

戻り値の型は、関数が呼ばれた場所に戻す値の型を示す。仮引数宣言は、関数が受け取るデータを格納するための変数の宣言である。

また、関数を呼び出す前に、あらかじめ関数の宣言（プロトタイプ宣言）をする。

[書式]

```
戻り値の型 関数名(仮引数型, ……);
```

[例]

```
int d_max(int, int);
```

[2] ポインタ

コンピュータでプログラムや変数を記憶しているメモリには、アドレスと呼ばれる記憶場所を示すメモリ番地が割り振られている。C言語では、これをポインタと呼ばれる方法で利用することにより、アセンブラー言語に近いプログラムを記述することができる。

[例]

右の図の場合、変数bの内容は、3.14
変数bのアドレスは、FF4番地

このアドレスを記憶している変数は、b_ad

①アドレス演算子

変数のアドレスを表す演算子。

[例]

変数bのアドレスを表す。 $\Rightarrow \&b$
&bの値はFF4になる（変数bのアドレスはFF4番地）。

②間接演算子

ポインタ変数が示すメモリの内容を表す演算子。

[例]

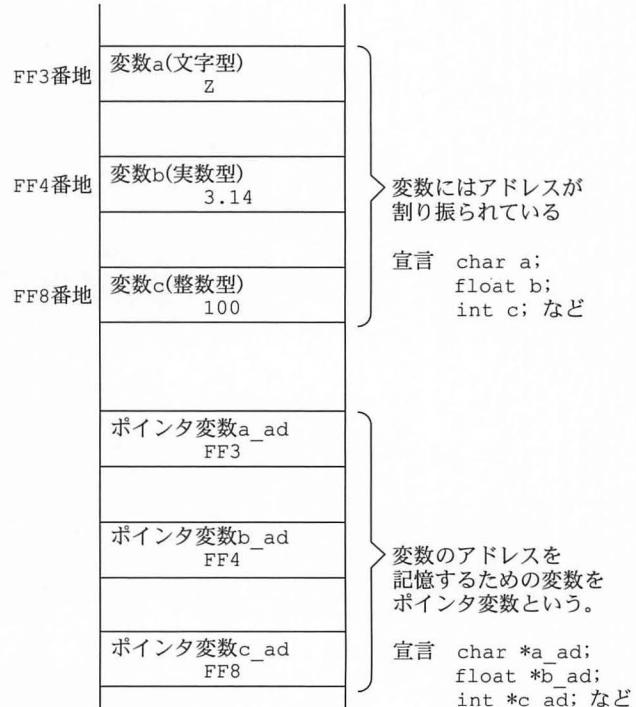
ポインタ変数b_adが示すメモリの内容を表す。
 $\Rightarrow *b_{ad}$
したがって、*b_adの値は3.14になる（b_adの値はFF4なので、FF4番地の内容3.14）。

③ポインタ変数の宣言

ポインタ変数が示すメモリの内容のデータ型で宣言する。

[例]

```
float *b_ad; (b_adの内容がアドレス（整数型）でもint b_ad;とはしない)  
ポインタ変数名b_adに*を付けてポインタ変数であることを明示する。  
*b_adは、この場合3.14（実数型）なのでfloatで宣言する。
```



④配列とポインタ

配列はコンピュータのメモリの連続する領域を使用することから、ポインタを用いると任意の配列要素を参照できる。

[例]

```
:  
char moji[6] = "JAPAN";  
char *moji_p;  
moji_p = moji;
```

moji_p = moji は moji_p = &moji[0]と同じ結果となる。すなわち、moji[0]のアドレスを、ポインタ変数moji_pに格納して利用できる。

□例題9.1□

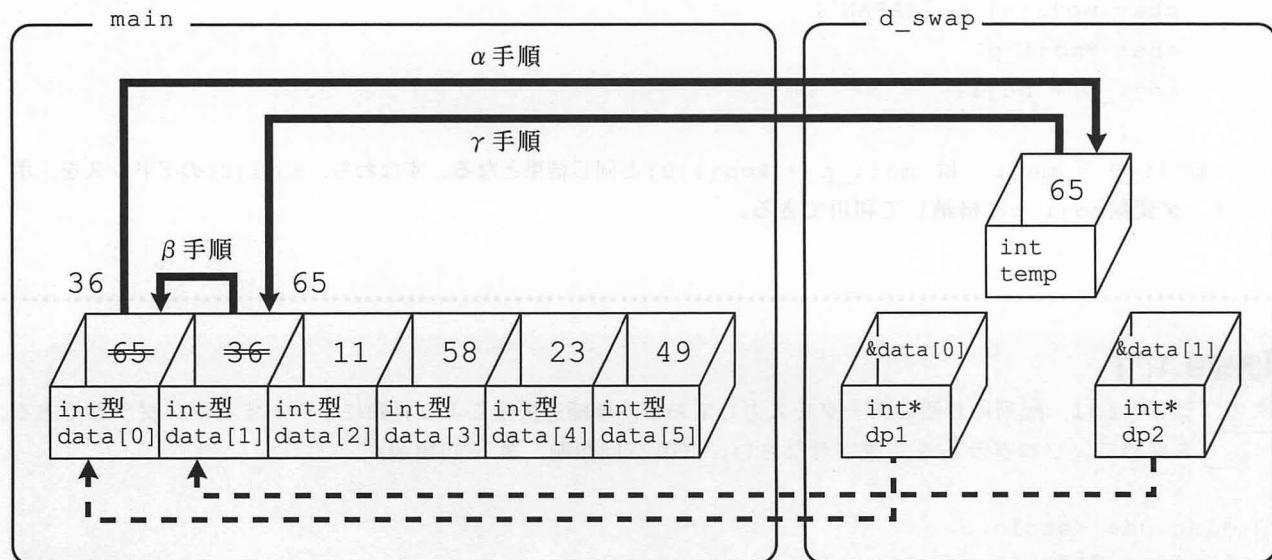
次のプログラムは、配列に6個のデータを入力しておき、単純選択法により昇順にソートするプログラムである。

_____を埋めて、プログラムを完成させなさい。（第27回類題、第28回類題）

```
1 #include <stdio.h>  
2 #define MAXN 6  
3 void d_swap(int *, int *);  
4  
5 int main(void)  
6 {  
7     int data[MAXN] = {65, 36, 11, 58, 23, 49};  
8     int i, j;  
9  
10    for (i = 0; i < [①]; i++) {  
11        for (j = [②]; j < MAXN; j++) {  
12            if (data[i] > data[j]) {  
13                d_swap(&data[i], &data[j]);  
14            }  
15        }  
16    }  
17  
18    for (i = 0; i < MAXN; i++) {  
19        printf("%5d", data[i]);  
20    }  
21    printf("\n");  
22  
23    return 0;  
24}  
25  
26 void d_swap(int *dp1, int *dp2)  
27 {  
28     int temp;  
29  
30     temp = *dp1;  
31     [③] = [④];  
32     *dp2 = temp;  
33 }
```

解説

①, ②は典型的な単純選択法の質問であるので、困難は無いと思う。③, ④については、`&data[0]`, `&data[1]`を引数として、`d_swap`関数が呼び出されたと仮定して図示してみると、次のようにになり、このタイミングでデータの入れ替えが行われるわけだが、データの交換手順を α , β , γ と名付けると、30行目が α 手順、31行目が β 手順、32行目が γ 手順に相当し、ポインタ`dp1`とポインタ`dp2`を利用しての操作は、`*dp1 = *dp2`であることが推測される。



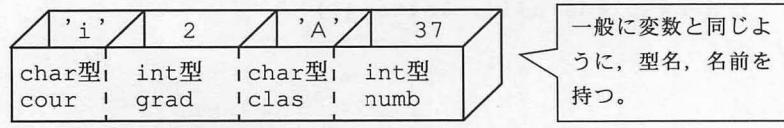
解答

- ①MAXN - 1 ②i + 1 ③*dp1 ④*dp2

要 点

[1] 構造体

構造体は、さまざまな既存のデータ型を組み合わせて作る新たなデータ型のようなものである。プログラムによっては、学科、学年、組、出席番号を「i科2年A組37番」のように組み合わせて、ひとまとめにして「生徒ID型」として取り扱うことができれば大変便利である。



(1つの大きな箱の中に間仕切りがあるイメージを持つ)

①構造体の定義と宣言

構造体を利用するには、構造体の型を定義（構造体の設計図に相当）し、構造体変数を宣言（構造体を作り、名前をつける）をしなければならない。左の書式は、定義と宣言を同時にした例であり、右は定義と宣言を分離した場合の書式である。いずれも同じ内容のプログラムであり、特に右のプログラム例の最後の行では、従来の変数宣言と比較すれば、`struct ID`が`int`などのデータ型で、`id2537`, `id3537`, `id3611`がプログラムで使用する変数名に対応することがわかる。

〔書式〕 定義と宣言を同時にした場合

```
struct 構造体の型名{  
    構造体のメンバのデータ型 メンバ名;  
    構造体のメンバのデータ型 メンバ名;  
    :  
} 構造体変数名, ……;
```

〔書式〕 定義と宣言を分離した場合

```
struct 構造体の型名{  
    構造体のメンバのデータ型 メンバ名;  
    構造体のメンバのデータ型 メンバ名;  
    :  
};  
struct 構造体の型名 構造体変数名, ……;
```

〔例〕

```
/*構造体の定義と宣言*/
struct ID{
    char cour;
    int grad;
    char clas;
    int numb;
} id2537, id3537, id3611;
```

〔例〕

```
/*構造体の定義*/
struct ID{
    char cour;
    int grad;
    char clas;
    int numb;
};

/*構造体変数の宣言*/
struct ID id2537, id3537, id3611;
```

②構造体の利用

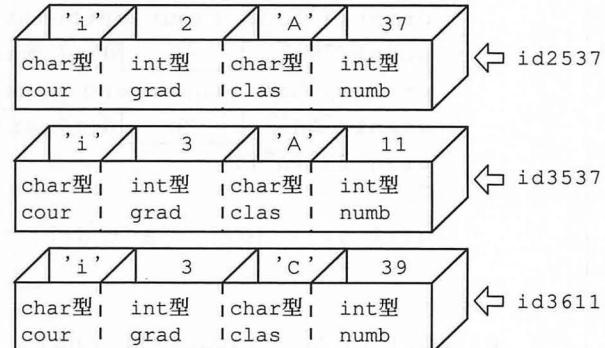
構造体変数も一般と変数や配列と同じく初期化宣言ができる。また、代入も行えるが、同じ型の構造体どうしで全データの代入の場合と、構造体の一部に対する代入の場合の2種類があり、前者は従来の変数と変わらない記述であるが、後者は定義のときに名付けたメンバ名を利用する。記述法は「.（ドット演算子）」を用いて「構造体変数名.メンバ名」となる。

〔書式〕 初期化宣言

```
struct 構造体の型名 構造体変数名 = {各メンバの初期値, ...};
```

〔例〕

```
struct ID id2537 = {'i', 2, 'A', 37};
struct ID id3537 = {'i', 3, 'A', 11};
struct ID id3611 = {'i', 3, 'C', 39};
```



〔書式〕 同じ型の構造体どうしの代入

構造体変数名 = 構造体変数名;

〔例〕

```
id2537 = id3537;
```

〔書式〕 構造体のメンバへの代入

構造体変数名.メンバ名 = 構造体変数名.メンバ名;

〔例〕

```
id3611.clas = 'C';
```

③構造体へのポインタ

構造体を指すポインタについても、構造体を取り扱うときは一般的のポインタと同じ取り扱いができる。ただし、ポインタが指して構造体のメンバを取り扱うときは、「->（アロー演算子）」を用いて「構造体ポインタ変数名 -> メンバ名」と記述することが多い。

〔書式〕 構造体のポインタの宣言

```
struct 構造体の型名 *ポインタ変数名, ...;
```

〔例〕

```
struct ID *ip1, *ip2, *ip3;
```

〔書式〕 構造体変数のアドレス

&構造体変数名

〔例〕

```
struct ID *ip1;
ip1 = &id2537;
```

〔書式〕 構造体メンバへのポインタ

構造体ポインタ変数名 -> メンバ名

〔例〕

```
ip2 -> numb = ip3 -> numb;
ip3 -> clas = 'C';
```

□例題9.2□

次のプログラムは、10人分の国語、数学、英語の点数（0～100）を入力し、各生徒の平均点を表示するものである。 [] を埋めて、プログラムを完成させなさい。

```
1 #include <stdio.h>
2 struct SCORE{ /* 構造体の定義 */
3     ① ja;
4     ① ma;
5     ① en;
6     double av;
7 };
8
9 int main(void)
10 {
11     ② d[10]; /* 構造体変数の宣言 */
12     int i;
13
14     for (i = 0; i < 10; i++) {
15         printf("%3d: Input japanese: ", i + 1);
16         scanf("%d", ③ d[i].ja);
17         printf("%3d: Input mathematics: ", i + 1);
18         scanf("%d", ③ d[i].ma);
19         printf("%3d: Input english: ", i + 1);
20         scanf("%d", ③ d[i].en);
21         printf("\n");
22
23         d[i].av = (d[i].ja + d[i].ma + d[i].en) / 3.0;
24     }
25
26     printf("\n No  japa  math  engl    ave\n");
27     for (i = 0; i < 10; i++) {
28         printf("%3d:%5d%5d%5d%7.1f\n", i + 1, d[i].ja, d[i].ma, d[i].en, ④ );
29     }
30
31     return 0;
32 }
```

解説

①は0～100までの点数となるので整数型を指定する。②は後ろのプログラムからja, ma, en, avをメンバに持つ構造体の型であることが判る。③については、構造体のメンバは基本的には従来の変数と同様に取り扱えよう。また、④は、%7.1fから実数である構造体のメンバを指定する。

解答

①int ②struct SCORE ③& ④d[i].av

[1] ファイルの取り扱い

C言語では、OSに関係なくファイルからの入力や出力などのファイル処理を行うための関数が用意されている。ファイル処理を行うためには、次の手順で処理を行う必要がある。

まず、ファイルポインタを宣言した後、ファイル処理の対象のファイルを、読んだり書いたりできるようにオープンする。

オープンできれば、実際にファイルの内容を処理する。最後に、ファイルをクローズして処理を終了する。

① ファイルのオープン

[書式] ファイルポインタ = fopen("ファイル名", "モード");

※モードについて "r": 読み込み "w": 書き込み "a": 追加

[例] fp = fopen("test.txt", "r");

ファイル名で示すファイルを、指定したモードの状態でオープンする。オープンに成功するとファイルを示すファイルポインタを返す。ファイルがオープンできなかった場合はNULLを返す。

② 文字列のファイルへの入力

[書式] fgets(文字配列名, 最大文字数, ファイルポインタ);

[例] fgets(buff, 16, fp);

ファイルポインタが示すファイルから文字列を配列に読み込む。最大文字数 - 1 個の文字数か改行文字を読み込んだ時点で読み込みを終了する。ファイルエンドかエラーの場合はNULLを返す。

③ 文字列のファイルへの出力

[書式] fputs(文字配列名, ファイルポインタ);

[例] fputs(buff, fp);

ファイルポインタが示すファイルへ配列から文字列を書き込む。エラーが発生するとEOFを返す。

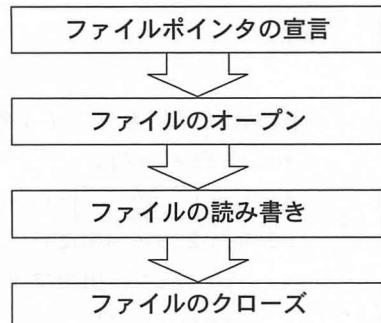
※入出力に関しては、ほかにも多くの関数が存在する。

④ ファイルのクローズ

[書式] fclose(ファイルポインタ);

[例] fclose(fp);

ファイルポインタで示すファイルをクローズする。そのときバッファはフラッシュされる。



□例題9.3□

次のプログラムは、C言語のソースファイルなどのテキストファイルを読み込み、出力例のように行の先頭に5桁の行番号を右詰で付けて、別のテキストファイルに出力するプログラムである。ただし、入力ファイル名および出力ファイル名はキーボードから入力するものとする。また、読み込むファイルは、1行が半角で換算して255文字以内であるものとする。 [] を埋めて、プログラムを完成しなさい。

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     [①] *fp1, *fp2; /* ファイル構造体へのポインタ */
6     char fname1[128], fname2[128], buf[256];
7     int no = 1;          /* 行番号 */
8
9     printf("入力ファイル名=");
10    gets(fname1);
11    fp1 = [②] (fname1, "r"); /* 入力ファイルを読み込みモード(read)で開く */
12    if (fp1 == NULL) {        /* ファイルがオープンできなかったとき */

```

```

13     printf("入力ファイルをオープンできません\n");
14     return 1; /* 0以外の数をOSに返して処理を終わる */
15 }
16
17 printf("出力ファイル名=");
18 gets(fname2);
19 fp2 = ② (fname2, "w"); /* 出力ファイルを書き込みモード(write)で開く */
20 if (fp2 == NULL) {
21     printf("出力ファイルをオープンできません\n");
22     return 1;
23 }
24
25 ③ (fgets(buf, 256, fp1) != NULL) { /* データを行単位で読み込み */
26     fprintf(fp2, "%05d: %s", no, buf); /* 行番号を付けて書き込み */
27     ④;
28 }
29
30 fclose(fp1); /* 入力ファイルのクローズ */
31 fclose(fp2); /* 出力ファイルのクローズ */
32
33 return 0;
34 }

```

解説

ファイル処理を行うためには、FILE構造体と呼ばれるファイル処理を行うための構造体を利用する。FILE構造体はstdio.hのなかで定義されている。また、FILEは大文字なので注意すること。

ファイルをオープンする場合、対象のファイルをオープンできなければエラーになる。ファイルをオープンする fopen 関数は、ファイルをオープンできなかったときはNULLを返す。上のプログラムでは、ファイルのオープンとエラーの確認を分けて行っているが、以下のようにまとめて行う記述をすることもある。

```

printf("入力ファイル名=");
gets(fname1);
if ((fp1 = fopen(fname1, "r")) == NULL) {
    printf("入力ファイルがオープンできません\n");
    return 1;
}

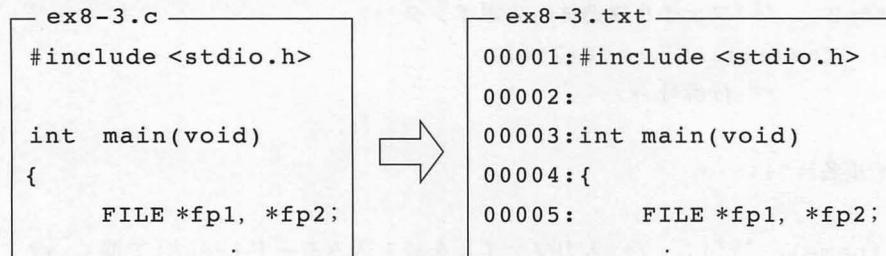
```

解答

①FILE ②fopen ③while ④no++

《出力例》

入力ファイル名=ex8-3.c
出力ファイル名=ex8-3.txt



練習問題

- 1 次のプログラムは正の10進数を2進数に変換するプログラムである。□を埋めて、プログラムを完成させなさい。（第36回）

```

1 #include <stdio.h>
2 void binary(int);
3
4 int main(void)
5 {
6     int nu;
7
8     printf("Input decimal :");
9     scanf("%d", &nu);
10
11    printf("binary : ");
12    if (nu [①] 0) {
13        binary(nu);
14        printf("\n");
15    }
16    else if (nu [②] 0) {
17        printf("0\n");
18    }
19    else {
20        printf("Input error\n");
21    }
22
23    return 0;
24 }
25
26 void binary(int n)
27 {
28     int mod, dvi;
29
30     if (n > 0) {
31         mod = n [③] 2;
32         dvi = n / 2;
33         binary([④]);
34         printf("%d", mod);
35     }
36 }
```

商が0になれば終わり

$$\begin{array}{r} 2) \underline{10} \\ 2) \underline{5} \cdots 0 \\ 2) \underline{2} \cdots 1 \\ 2) \underline{1} \cdots 0 \\ 0 \cdots 1 \end{array}$$

余りを下から読む

main()関数より
binary(10)を呼び出す ↓ ↑ 0を表示

$$\begin{array}{l} 10 > 0 \\ \text{mod} = 10 \% 2 = 0 \\ \text{dvi} = 10 / 2 = 5 \end{array}$$

binary(5)を呼び出す ↓ ↑ 1を表示

$$\begin{array}{l} 5 > 0 \\ \text{mod} = 5 \% 2 = 1 \\ \text{dvi} = 5 / 2 = 2 \end{array}$$

binary(2)を呼び出す ↓ ↑ 0を表示

$$\begin{array}{l} 2 > 0 \\ \text{mod} = 2 \% 2 = 0 \\ \text{dvi} = 2 / 2 = 1 \end{array}$$

binary(1)を呼び出す ↓ ↑ 1を表示

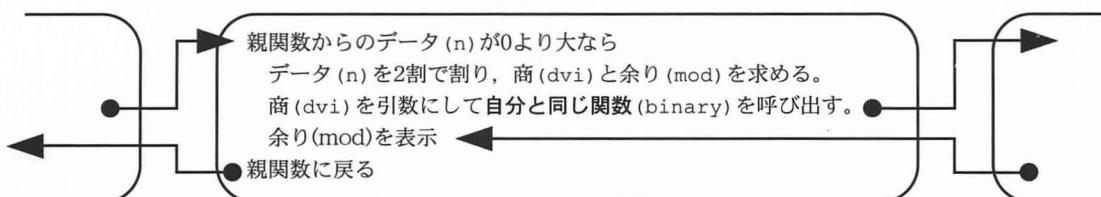
$$\begin{array}{l} 1 > 0 \\ \text{mod} = 1 \% 2 = 1 \\ \text{dvi} = 1 / 2 = 0 \end{array}$$

binary(0)を呼び出す ↓ ↑ 何もない

$$\begin{array}{l} 0 > 0 \\ \text{mod} = 0 \% 2 = 0 \\ \text{dvi} = 0 / 2 = 0 \end{array}$$

ヒント

上のプログラムでは、関数の中から、自分と同じ関数を呼び出す再帰呼び出しを使用している。コピーはコンピュータにとり、得意技なのでコンピュータらしい技法と言える。



- 2 次のプログラムは、与えられたデータの個数と合計を求めて表示するものである。□を埋めて、プログラムを完成しなさい。ただし、データの終わりは -99 で判定する。（第32回、第37回）

```
1 #include <stdio.h>
2 int data_su(int *);
3 int goukei(int *);
4
5 int main(void)
6 {
7     int a[] = {89, 56, 23, 80, 46, -99};
8     int su, kei;
9
10    su = data_su(a);
11    printf("データ数は %d\n", su);
12
13    kei = goukei(a);
14    printf("合計は %d\n", kei);
15
16    return 0;
17 }
18
19 int data_su(int *a)
20 {
21     int i;
22
23     i = 0;
24     while (□①) {
25         i++;
26         □②;
27     }
28
29     return i;
30 }
31
32 int goukei(□③)
33 {
34     int wa = 0;
35
36     while (*a != -99) {
37         wa = □④;
38         □⑤;
39     }
40
41     return wa;
42 }
```

- 3 次のプログラムは、与えられた文字列の文字数を求め、さらにアルファベット小文字をアルファベット大文字に変換して表示するものである。[] を埋めて、プログラムを完成しなさい。（第33回、第35回類題）

```
1 #include <stdio.h>
2 int mojisu(char *);
3 void henkan(char *);
4
5 int main(void)
6 {
7     char moji[] = "Tokyo Osaka Nagoya";
8     int nagasa;
9
10    printf("文字列は %s\n", moji);
11
12    nagasa = [①];
13    printf("文字数は %d\n", nagasa);
14
15    henkan(moji);
16    printf("変換後は %s\n", moji);
17
18    return 0;
19}
20
21 int mojisu(char *m)
22 {
23     int i;
24
25     i = 0;
26     while (*m [②] '\0') {
27         i++;
28         m++;
29     }
30
31     return i;
32 }
33
34 void henkan(char *m)
35 {
36     while (*m [③] '\0') {
37         if ((*m >= 'a') [④] (*m <= 'z')) {
38             *m = *m - ([⑤]);
39         }
40         m++;
41     }
42 }
```

- 4 次のプログラムは、文字列と文字列を比較し、一致していれば1を、一致していなければ0を表示するものである。□を埋めて、プログラムを完成しなさい。（第34回）

```
1 #include <stdio.h>
2 int hikaku(char *, char *);
3
4 int main(void)
5 {
6     char moji1[] = "Tokyo";
7     char moji2[] = "Tokyo";
8     char moji3[] = "Osaka";
9     char moji4[] = "osaka";
10    int kekka;
11
12    kekka = hikaku(moji1, moji2);
13    printf("文字列比較の結果は %d\n", kekka);
14
15    kekka = hikaku(moji3, moji4);
16    printf("文字列比較の結果は %d\n", kekka);
17
18    return 0;
19 }
20
21 int hikaku(char *m1, char *m2)
22 {
23     while (*m1 == □①□ *m2) {
24         if (*m1 == □②□) {
25             □③□;
26         }
27         □④□;
28         m2++;
29     }
30
31     □⑤□;
32 }
```

- 5 次のプログラムは、与えられた文字列のアルファベット小文字をアルファベット大文字に変換し、さらに、空白部分を削除してつめて表示するものである。[] を埋めて、プログラムを完成しなさい。
(第27回類題、第28回類題)

```
1 #include <stdio.h>
2 void henkan(char *);
3 void sakujo(char *);
4
5 int main(void)
6 {
7     char moji[] = "Japan & Korea";
8
9     printf("変換前の文字列は %s\n", moji);
10
11    henkan(moji);
12    printf("変換後の文字列は %s\n", moji);
13
14    sakujo(moji);
15    printf("削除後の文字列は %s\n", moji);
16
17    return 0;
18 }
19
20 void henkan(char *m)
21 {
22     while (*m != '\0') {
23         if ((*m >= 'a') [①] (*m <= 'z')) {
24             *m = *m - ([②]);
25         }
26         [③];
27     }
28 }
29
30 void sakujo(char *m)
31 {
32     int i;
33
34     while (*m != '\0') {
35         if (*m [④] ', ') {
36             i = 0;
37             while (*(m + i) != '\0') {
38                 *(m + i) = *( [⑤]);
39                 i++;
40             }
41         }
42         else {
43             m++;
44         }
45     }
46 }
```

- 6 次のプログラムは、与えられた複数の文字列を昇順に表示するものである。□を埋めて、プログラムを完成しなさい。

```
1 #include <stdio.h>
2 #include <string.h>
3 void disp(char *[]);
4 void sort(char *[]);
5 void swap(char [], int, int);
6
7 int main(void)
8 {
9     char *country[5] = {"Japan",
10         "Korea",
11         "China",
12         "Jamaica",
13         "India"};
14
15     ①;
16     disp(country);
17
18     return 0;
19 }
20
21 void sort(char *c[])
22 {
23     int i, j;
24
25     for (i = 0; i < 4; i++) {
26         for (j = i + 1; j < 5; j++) {
27             if (strcmp(c[i], c[j]) > 0) {
28                 ②;
29             }
30         }
31     }
32 }
33
34 void swap(char ③, int i, int j)
35 {
36     char ④;
37
38     temp = c[i];
39     c[i] = c[j];
40     c[j] = temp;
41 }
42
43 void disp(char *c[])
44 {
45     int i;
46
47     for (i = 0; i < 5; i++) {
48         printf("%s\n", ⑤);
49     }
50 }
```

- 7 次のプログラムは、`maze[0][0]`からスタートし、「n」、「e」、「w」、「s」のコマンドを打ち、`maze[4][4]`のゴールめざし迷路の中を探検するプログラムである。を埋めて、プログラムを完成させなさい。

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     struct NEWS {
6         int n;
7         int e;
8         int w;
9         int s;
10    } maze[5][5] = {{0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, -4, 1, 0},
11                  {1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1},
12                  {1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1},
13                  {1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0},
14                  {1, 0, 0, -4, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0}};
15    char com[256];
16    int y = 0, [①];
17
18    while (x != 4 [②] y != 4) {
19        printf("Input Command: ");
20        scanf("%s", com);
21
22        switch (com[0]) {
23            case 'n':
24                if (maze[y][x].n == 0) {
25                    printf("You knock against the wall.\n");
26                }
27                else {
28                    y = y - maze[y][x].n;
29                }
30                break;
31            case 'e':
32                if (maze[y][x].e == 0) {
33                    printf("You knock against the wall.\n");
34                }
35                else {
36                    x = x [③] maze[y][x].e;
37                }
38                break;
39            case 'w':
40                if (maze[y][x].w == 0) {
41                    printf("You knock against the wall.\n");
42                }
43                else {
44                    x = x [④] maze[y][x].w;
45                }
46                break;
47            case 's':
48                if (maze[y][x].s == 0) {
49                    printf("You knock against the wall.\n");
50                }
51                else {
52                    y = y + maze[y][x].s;
53                }
54                break;
55            }
56        printf("maze[%d] [%d]\n", y, x);
57    }
58    printf("Bingo!! You are exit\n");
59
60    return 0;
61 }

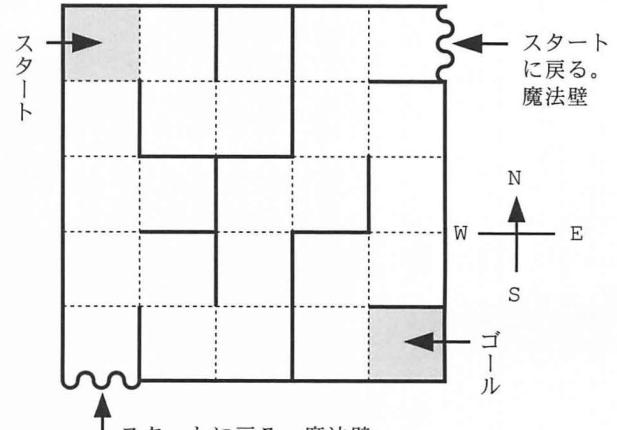
```

ヒント

構造体の値は、北・東・西・南への移動量を示す。
例えば、スタートエリアmaze[0][0]は{0, 1, 0, 1}である。

ヒント

構造体の値は、北・東・西・南への移動量を示す。
例えば、スタートエリア `maze[0][0]` は `{0, 1, 0, 1}` である。

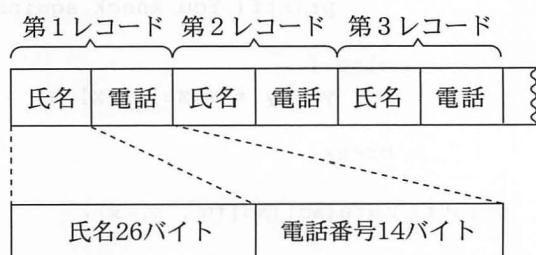


- 8 次のプログラムは、下図のようなレコード形式で保存されている「data.dat」という名前のデータファイルにデータを追加するプログラムである。ただし、ファイルが存在しなければ、ファイルを作成するものとする。を埋めて、プログラムを完成させなさい。なお、氏名、電話番号ともデータは左詰で記録するものとする。

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     FILE  ① ;
6     char name[26];
7     char tel[14];
8
9     fp = fopen("data.dat", "a");
10    if (fp ==  ② ) {
11        printf("¥aファイルをオープンできません！¥n");
12        return 1;
13    }
14
15    printf("CTRL+Z で入力を終了します¥n");
16    while (1) {
17        printf("氏名 = ");
18        if ( ③ (name) == NULL) {
19            break;
20        }
21
22        printf("電話番号 = ");
23        gets(tel);
24        fprintf( ④ , "%-26s%-14s", name, tel);
25    }
26
27     ⑤ (fp);
28
29    return 0;
30 }
```

レコード形式



- 9 次のプログラムは、問題 8 のデータファイル「data.dat」から、レコードの番号を指定してデータを読み出して表示するものである。[] を埋めて、プログラムを完成させなさい。なお、氏名、電話番号ともデータは左詰で記録されており、残りの部分は1つ以上の空白で占められている。（第29回）

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     FILE *fp;
6     char name[26];
7     char tel[14];
8     int n = 1;
9
10    if ((fp = [ ] ("data.dat", "r")) == NULL)
11    {
12        printf("ファイルをオープンできません！\n");
13        return 1;
14    }
15
16    while (1)
17    {
18        printf("データ番号は？（終了=0）");
19        scanf("%d", &n);
20        if (n [ ] 0)
21        { /* 終了条件チェック */
22            break;
23        }
24
25        fseek([ ], (n - 1) * 40L, [ ]); /* ポインタの移動 */
26        if ([ ](fp, "%26s%14s", name, tel) == EOF)
27        {
28            printf("その番号のデータはありません！\n");
29        }
30        else {
31            printf("%-26s %-14s\n", name, tel);
32        }
33    }
34
35    fclose(fp);
36
37    return 0;
38 }
```

9.2 各種アルゴリズム

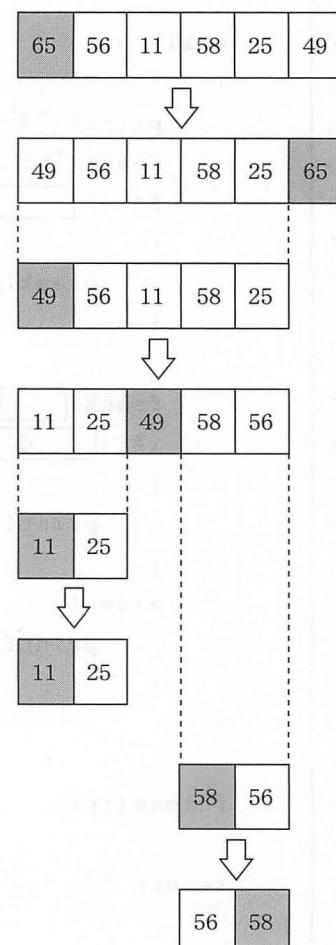
練習問題

- 10 次のプログラムは、配列に6個のデータを入力しておき、クイックソート法により昇順にソートするプログラムである。 [] を埋めて、プログラムを完成させなさい。

```
1 #include <stdio.h>
2 #define MAXN 6
3 void d_print(int []);
4 void qui_sort(int [], int, int);
5 void d_swap(int *, int *);
6
7 int main(void)
8 {
9     int d[MAXN] = {65, 56, 11, 58, 25, 49};
10
11    d_print(d);
12    qui_sort(d, 0, MAXN - 1);
13    d_print(d);
14
15    return 0;
16 }
17
18 void qui_sort(int d[], int left, int right)
19 {
20     int lp, rp, mid;
21
22     [①] = d[left];
23     lp = left;
24     rp = right;
25
26     while (lp < rp) {
27         while (d[lp] <= mid && lp <= rp) {
28             [②];
29         }
30         while (mid < d[rp] && lp <= rp) {
31             [③];
32         }
33         if (lp < rp) {
34             d_swap(&d[lp], &d[rp]);
35         }
36     }
37     d_swap(&d[left], &d[rp]);
38
39     if (left < rp - 1) {
40         qui_sort(d, left, [④]);
41     }
42     if (rp + 1 < right) {
```

ヒント

クイックソート法の昇順では、ある分割要素を選び、この値以上のグループと未満のグループに分け、1つの要素になるまでグループの分割を繰り返しソートしていくものである。次の図では、ソート範囲の一番左の値を分割要素としている。



```

43         qui_sort(d, [REDACTED], right);
44     }
45 }
46
47 void d_swap(int *buf1, int *buf2)
48 {
49     int work;
50
51     work = *buf1;
52     *buf1 = *buf2;
53     *buf2 = work;
54 }
55
56 void d_print(int d[])
57 {
58     int i;
59
60     for (i = 0; i < MAXN; i++) {
61         printf("%5d", d[i]);
62     }
63     printf("\n");
64 }
```

- 11 次のプログラムは、配列に6個のデータを入力しておき、ヒープソート法により昇順にソートするプログラムである。[REDACTED]を埋めて、プログラムを完成させなさい。

```

1 #include <stdio.h>
2 #define MAXN 7
3 void d_print(int []);
4 void heap_sort(int [], int);
5 void heap_build(int [], int, int);
6 void d_swap(int *, int *);
7
8 int main(void)
9 {
10     int data[MAXN] = {0, 65, 56, 11, 58, 25, 49}; /* data[0]の0はダミー */
11
12     d_print(data);
13     heap_sort(data, MAXN - 1);
14     d_print(data);
15
16     return 0;
17 }
18
19 void heap_sort(int d[], int n)
20 {
21     int i;
22
23     for (i = n / 2; i >= 1; i--) {
```

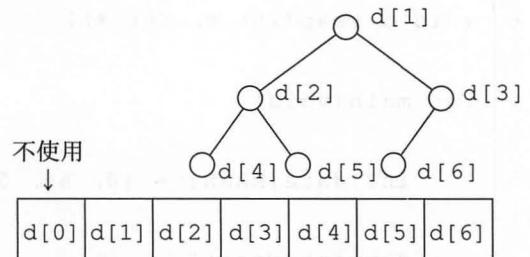
```

24         heap_build(d, i, n);
25     }
26
27     while (n > 1) {
28         d_swap(&d[①], &d[n]);
29         n--;
30
31         heap_build(d, 1, n);
32     }
33 }
34
35 void heap_build(int d[], int pa, int n)
36 {
37     int ch;
38
39     ch = 2 * pa; /* 左の子 */
40     while (ch <= n) {
41         if (ch < n && d[②] < d[③]) { /* 左右の子の内、大きい子 */
42             ch++;
43         }
44
45         if (d[pa] >= d[ch]) {
46             break;
47         }
48
49         d_swap(&d[pa], &d[ch]);
50         pa = ④; /* 次の親 */
51         ch = ⑤; /* 次の左の子 */
52     }
53 }
54
55 void d_swap(int *buf1, int *buf2)
56 {
57     int work;
58
59     work = *buf1;
60     *buf1 = *buf2;
61     *buf2 = work;
62 }
63
64 void d_print(int d[])
65 {
66     int i;
67
68     for (i = 1; i < MAXN; i++) {
69         printf("%5d", d[i]);
70     }
71     printf("\n");
72 }

```

ヒント

完全2分木で、どの親と子を取り出しても、親のデータ>子のデータとなっている2分木を作り、最上位の親（根）を取り出した後、残りのデータで2分木を整理する。この操作を繰り返し、取り出したデータを順に並べることをヒープソートと言う。配列で2分木を作ると、親と子の添え字の間には次のような関係ができる。



親の添え字 * 2 → 左の子の添え字

親の添え字 * 2 + 1 → 右の子の添え字

子の添え字 / 2 → 親の添え字

- 12 1とその数以外では割り切れない数を素数という。次のプログラムは、エラトステネスのふるいというアルゴリズムを利用して、キーボードから入力した整数n以下のすべての素数を表示するプログラムである。
_____を埋めて、プログラムを完成させなさい。ただし、nが1000以上の時は、再入力させるものとする。（第34回）

```
1 #include <stdio.h>
2 #define MAX 1000
3
4 int main(void)
5 {
6     int j, k, n, cnt, s[MAX];
7
8     do {
9         printf("1000未満の数 = ");
10        scanf("%d", &n);
11    }while (n < 1 _____ n >= MAX);
12
13    s[1] = 0;
14    for (j = 2; j <= n; j++) {
15        s[j] = 1;
16    }
17    for (j = 2; j * j <= n; j++) {
18        if (s[j] != 0) {
19            for (k = 2 * j; k <= n; k += j) {
20                s[k] = _____;
21            }
22        }
23    }
24
25    cnt = _____;
26    for (j = 2; j <= n; j++) {
27        if (s[j] _____ 0) {
28            printf("%7d", j);
29            cnt++;
30            if (cnt % 10 == 0) {
31                printf("\n");
32            }
33        }
34    }
35
36    printf("%n%d以下の素数の数は、%d個です。%n", n, cnt);
37
38    return 0;
39 }
```

ヒント

エラトステネスのふるいについては、問題集96ページ・11を参考。

13 次のプログラムは、マクローリン級数展開をもちいて、指数関数 e^x を求めて表示するプログラムである。

□を埋めて、プログラムを完成させなさい。（第35回）

```
1 #include <stdio.h>
2 #define EPS 1.0E-8 /* 相対打ち切り誤差 */
3 #define MAX 200
4 double exp_func(double);
5
6 int main(void)
7 {
8     double x, ex;
9
10    while (1) {
11        printf("Xを入力(0で終了)＼n");
12        scanf("%lf", &x);
13        if (x == 0) {
14            □①□;
15        }
16
17        ex = exp_func(x);
18        printf("exp(%f) = %e＼n", x, ex);
19    }
20
21    return 0;
22 }
23
24 double exp_func(double x)
25 {
26     int k;
27     double ax, d, s, t;
28
29     if (x > 0) {
30         ax = □②□;
31     }
32     else {
33         ax = -1 * x;
34     }
35
36     s = d = 1.0;
37     for (k = 1; k <= MAX; k++) {
38         t = s;
39         d = d * ax / k;
40         s = s + d;
41         if (d < EPS * t) {
42             break;
43         }
44     }
45
46     if (k > □③□) {
47         printf("収束しない！＼n");
48
49         return 0.0;
50     }
51     else if (x > 0) {
52         return s;
53     }
54     else {
55         return □④□;
56     }
57 }
```

ヒント

指数関数 e^x をマクローリン級数に展開すると、次のように表される。

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^{k-1}}{(k-1)!} + \frac{x^k}{k!} + \cdots$$

この級数は無限級数なので、 $k-1$ 項までの和を S_{k-1} 、 k 項までの和を S_k とするとき、 $|S_k - S_{k-1}|$ が十分に小さくなったら収束したと見なし、処理を終了する。

実際の処理では、 $\frac{|S_k - S_{k-1}|}{|S_{k-1}|} < \text{EPS}$

となる EPS を精度に応じて適当に設定する。このとき、 $|S_k - S_{k-1}|$ を打ち切り誤差、 $\frac{|S_k - S_{k-1}|}{|S_{k-1}|}$ を相対打ち切り誤差という。

問題集99ページ・問題 14 を参考。

- 14 次のプログラムは、キーボードから入力した正の実数 Q の立方根 $\sqrt[3]{Q}$ を、二分法を用いて求めるプログラムである。□を埋めて、プログラムを完成させなさい。（第32回）

```

1 #include <stdio.h>
2 #include <math.h>
3 #define EPS 1E-8
4 double c_func(double, double);
5
6 int main(void)
7 {
8     double a, b, c, q, dx, fa, fc;
9
10    do {
11        printf("Q=");
12        scanf("%lf", &q);
13    } while (q <= 0);
14
15    a = 1.0;
16    b = q;
17
18    do {
19        c = □①□ / 2.0;
20        fa = c_func(a, q);
21        fc = c_func(c, q);
22        dx = fabs(b - a);
23        if (fc == 0) {
24            break;
25        }
26        if (fa * fc < 0) {
27            □②□;
28        }
29        else {
30            □③□;
31        }
32    } while (dx > EPS);
33
34    printf("%fの立方根は, %f\n", q, c);
35
36    return 0;
37}
38
39 double c_func(double x, double q)
40 {
41     double y;
42
43     y = □④□;
44
45     return y;
46}

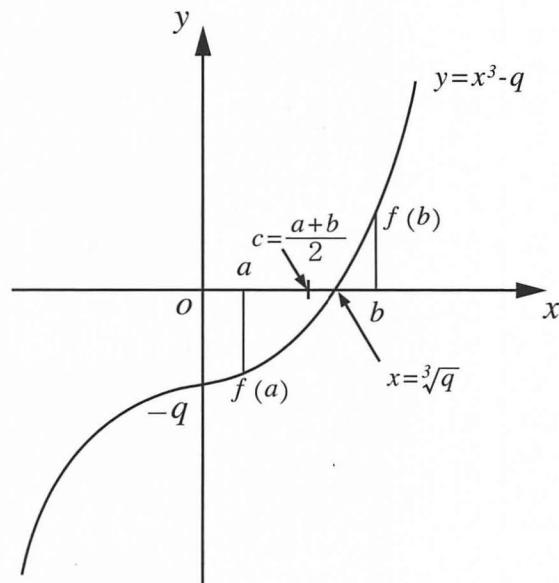
```

ヒント

実数 a の立方根は、方程式 $x^3 = q$ つまり $x^3 - q = 0$ の解として求められる。方程式 $f(x) = 0$ の解は、 $y = f(x)$ のグラフと x 軸との交点であるので、図のように解の前後で関数の値の符号は反転する。つまり、区間 $a < x < b$ において、 $f(a) \cdot f(b) < 0$ ならばこの区間の間に $f(x) = 0$ となる x が存在する。

そこで、区間の中点 $c = \frac{a+b}{2}$ をとり、範囲を $a < x \leq c$ と $c \leq x < b$ に二分する。ここでもしも $f(c) = 0$ なら c が解である。また、 $f(a)$ と $f(c)$ が異符号なら解は $a < x < c$ の区間にあり、 $f(c)$ と $f(b)$ が異符号なら解は $c < x < b$ の区間にある。解がある方の区間をさらに二分し、そのどちらかに解があるかを調べる。これを繰り返し、区間が十分小さくなったら、その区間の中央の値を解とする。

問題集104ページ・問題 17 を参考。



- 15 次のプログラムは、キーボードから入力した正の実数 Q の立方根 $\sqrt[3]{Q}$ を、ニュートン法を用いて求めるプログラムである。□を埋めて、プログラムを完成させなさい。

```

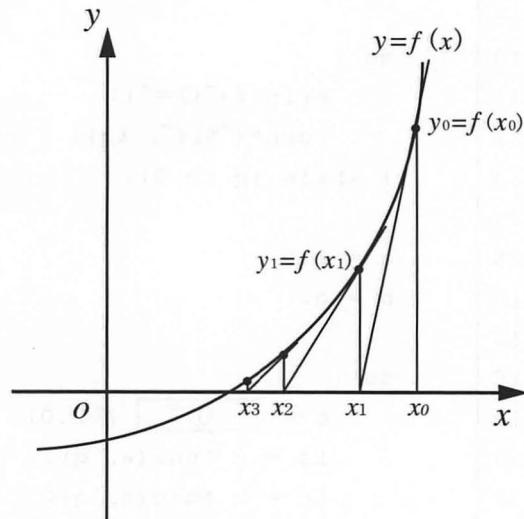
1 #include <stdio.h>
2 #include <math.h>
3 #define EPS 1E-8
4 #define MAX 50
5 double c_func(double, double);
6 double d_func(double);
7
8 int main(void)
9 {
10     int cnt = 0;
11     double a, b, fx, fd, dx, q;
12
13     do {
14         printf("Q=");
15         scanf("%lf", &q);
16     } while (q <= 0);
17
18     a = □①□;
19     while (cnt < MAX) {
20         fx = c_func(a, q);
21         fd = d_func(a);
22         b = a - □②□ / □③□;
23
24         dx = fabs(a - b);
25         if (dx < EPS) {
26             break;
27         }
28         else {
29             a = b;
30         }
31
32         □④□;
33     }
34
35     if (cnt < MAX) {
36         printf("%f の立方根は, %f\n", q, b);
37     }
38     else {
39         printf("収束しません\n");
40     }
41
42     return 0;
43 }
44
45 double c_func(double x, double q)
46 {
47     double y;
48
49     y = x * x * x - q;
50
51     return y;

```

ヒント

ニュートン法については、

問題集103ページ・例題7. 8を参考。



```

52 }
53
54 double d_func(double x)
55 {
56     double y;
57
58     y = 3.0 * x * x;
59
60     return y;
61 }

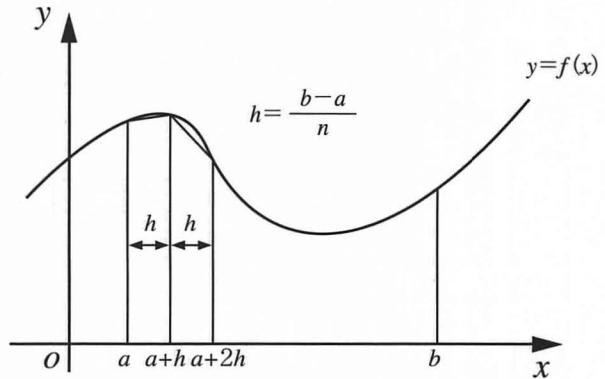
```

- 16 次のプログラムは、台形近似を利用して定積分 $\int_0^1 \frac{4}{x^2+1} dx$ の近似値を求めるプログラムである。
 ブランクを埋めて、プログラムを完成させなさい。(第36回)

```

1 #include <stdio.h>
2 #define f(x) 4.0 / (x * x + 1.0)
3
4 int main(void)
5 {
6     int kubun, k;
7     double x, h, y0, y1, menseki;
8
9     while (1) {
10         printf("分割数を入力(0で終了)\n");
11         scanf("%d", &kubun);
12
13         if (kubun == 0) { /* 終了条件 */
14             ①;
15         }
16         else if (kubun < 0) { /* 再入力 */
17             ②;
18         }
19
20         h = 1.0 / (double)kubun;
21         x = 0.0;
22         ③ = 0.0;
23
24         for (k = 0; k < kubun; k++) {
25             y0 = f(x);
26             x += ④;
27             y1 = f(x);
28             menseki += h * (y0 + y1) / 2.0;
29         }
30         printf("面積= %.10f\n", menseki);
31     }
32
33     return 0;
34 }

```



ヒント

定積分 $\int_a^b f(x)dx$ の値は、 $x=a, x=b, y=0, y=f(x)$ 、
 によって囲まれた部分の面積に等しい。

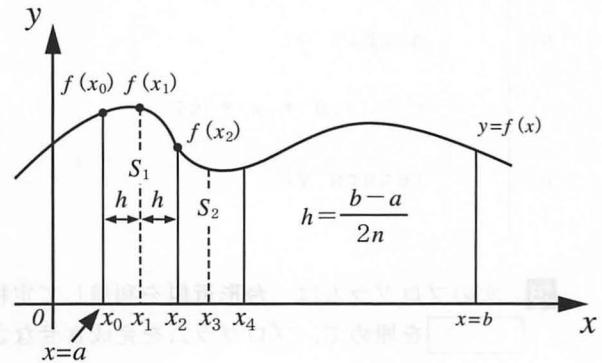
図のように、区間 $[a, b]$ を細かく n 分し、各
 区間の幅を h とすると、 $h = \frac{b-a}{n}$ となる。こ
 のとき、図のように各等分した点の関数の値を
 直線で結ぶと台形ができる。区間 $[a, b]$ の台形
 の面積を合計すれば、定積分の近似値が得られる。

問題集101ページ問題 15 を参考

- 17 次のプログラムは、シンプソン法を利用して定積分 $\int_0^1 \frac{4}{x^2+1} dx$ の近似値を求めるプログラムである。
 を埋めて、プログラムを完成させなさい。

```

1 #include <stdio.h>
2 #define f(x) 4.0 / (x * x + 1.0)
3
4 int main(void)
5 {
6     int n, kubun, k;
7     double x, y0, y1, y2, h, menseki;
8
9     while (1) {
10         printf("分割数を入力(0で終了) %n");
11         scanf("%d", &n);
12
13         if (n  0) { /* 終了条件 */
14             break;
15         }
16         else if (n  0) { /* 再入力 */
17             continue;
18         }
19
20         kubun =  ③;
21         h = 1.0 / (double)kubun;
22         x = 0.0;
23         menseki = 0.0;
24
25         for (k = 1; k < kubun; k += 2) {
26             y0 = f(x);
27             x += h;
28             y1 = f(x);
29             x += h;
30             y2 = f(x);
31
32             menseki  ④ h * (y0 + 4.0 * y1 + y2) / 3.0;
33         }
34
35         printf("面積= %.10f\n", menseki);
36     }
37
38     return 0;
39 }
```



ヒント

シンプソン法は、微小区間を二次曲線により曲線を近似して定積分の近似値を求める方法である。関数 $y=f(x)$ 上の 3 点 $f(x_0), f(x_1), f(x_2)$ を通る 2 次曲線を $y=g(x)$ とするとき、微小区間 $[x_0, x_2]$ における 2 次曲線で近似した面積 S_1 は、

$$S_1 = \int_{x_0}^{x_2} g(x) dx = \frac{h}{3} \{ f(x_0) + 4f(x_1) + f(x_2) \}$$
で求められる。これをシンプソンの公式という。

問題集100ページ例題7. 7を参考

- 18 次のプログラムは、モンテカルロ法により円周率πを求めるものである。[] を埋めて、プログラムを完成させなさい。（第26回）

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     double x, y, pi;
8     long i, a, b;
9
10    printf("乱数の個数 = ");
11    scanf("%ld", [①]);
12
13    srand((unsigned int)time(NULL)); /* 亂数の初期化 */
14    [②] = 0;
15    i = a;
16    while ([③]) {
17        x = (double)rand() / RAND_MAX; /* [0, 1] の範囲の乱数を発生 */
18        y = (double)rand() / RAND_MAX;
19
20        if ((y * y + x * x) <= 1) {
21            [④];
22        }
23    }
24
25    pi = 4.0 * (double)b / (double)a;
26    printf("π = %9.7f\n", pi);
27
28    return 0;
29 }
```

ヒント

プログラムのrand関数は、0からRAND_MAXまでの疑似乱数を発生させる関数である。RAND_MAXは「stdlib.h」の中でマクロ定義されている。乱数の発生系列を変更するにはsrand関数を用いる。

モンテカルロ法については、問題集95ページ 10 を参考。

19 縦・横・斜めの合計がすべて同じになるように、1～ N^2 までの連続した異なる数を、正方形に並べたものをN次の魔方陣という。

次のプログラムは、Nを入力したときに、N次の魔方陣（Nは15以下の奇数）を作成して表示するプログラムである。□を埋めて、プログラムを完成させなさい。（第33回）

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int mg[16][16], i, j, k, n;
6
7     while (1) {
8         printf("15以下の奇数を入力してください！\n");
9         scanf("%d", &n);
10        if (n % 2 && n >= 1 && n <= 15) {
11            break;
12        }
13    }
14
15    j = □①□ / 2;
16    i = 1;
17    mg[i][j] = 1;
18    for (k = 2; k <= □②□; k++) {
19        if ((k % n) == 1) {
20            i++;
21        }
22        else if (i == 1) {
23            i = □③□;
24            j++;
25        }
26        else if (j == n) {
27            i--;
28            j = □④□;
29        }
30        else {
31            i--;
32            j++;
33        }
34        mg[i][j] = k;
35    }
36
37    for (i = 1; i <= n; i++) {
38        for (j = 1; j <= n; j++) {
39            printf("%5d", mg[i][j]);
40        }
41        printf("\n");
42    }
43
44    return 0;
45 }
```

実行例

15以下の奇数を入力してください！

5	17	24	1	8	15
23	5	7	14	16	
4	6	13	20	22	
10	12	19	21	3	
11	18	25	2	9	

ヒント

問題集 97ページ 12 を参考

- 20 次のプログラムは、配列の文字列から、逐次探索法により指定した文字列を探索するプログラムである。
 ブランクを埋めて、プログラムを完成させなさい。 (第33回)

```

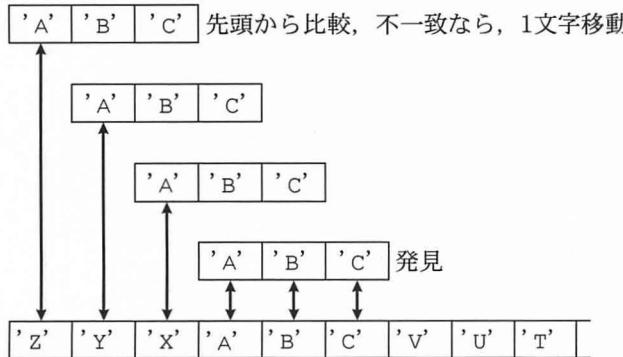
1 #include <stdio.h>
2
3 int main(void)
4 {
5     char data[] = "redundant arrays of inexpensive disk";
6     char pat[] = "disk";
7     char *dp, *pp;
8     int i;
9
10    dp = data;
11    pp = pat;
12    while (*dp [①] '\0') {
13        for (i = 0; *(dp + i) [②] *(pp + i) && *(dp + i) [③] '\0'; i++) {
14            }
15            if (*(pp + i) [④] '\0') {
16                break;
17            }
18
19            dp++;
20        }
21
22        if (*(pp + i) [⑤] '\0') {
23            printf("not ");
24        }
25        else {
26            printf("found\n");
27        }
28
29        return 0;
30    }

```

ヒント

逐次探索法：

探索パターン文字列の先頭から末尾に向かって、順番に文字の比較を行い、不一致になると先に進む。



- 21 次のプログラムは、0～100の点を入力し、上位5位までを表示するものである。□を埋めて、プログラムを完成させなさい。なお、負の値が入力されるとプログラムは終了する。

```
1 #include <stdio.h>
2 struct BEST {
3     int score;
4     struct BEST *next;
5 };
6
7 int main(void)
8 {
9     int iscore, i;
10    struct BEST rk[5];
11    struct BEST *top = &rk[0], *ins, *last, *op;
12
13    /* リスト連結部 */
14    for (i = 0; i < 4; i++) {
15        rk[i].score = 0;
16        rk[i].next = &rk[i + 1];
17    }
18    rk[4].score = 0;
19    rk[4].next = NULL;
20
21    /* データ入力部 */
22    printf("Input Score:");
23    scanf("%d", &iscore);
24    while (iscore >= 0) {
25        /* 検索部 */
26        op = top;
27        ins = NULL;
28        while (①) {
29            if (op->score < iscore && ins == NULL) {
30                ins = op;
31            }
32
33            if (op->next == NULL) {
34                last = op;
35            }
36
37            op = op->next;
38        }
39
40        /* 挿入部 */
41        if (ins != NULL) {
42            last->score = ②;
43
44            if (top != ins) {
45                op = top;
46                while (op->next != ins) {
```

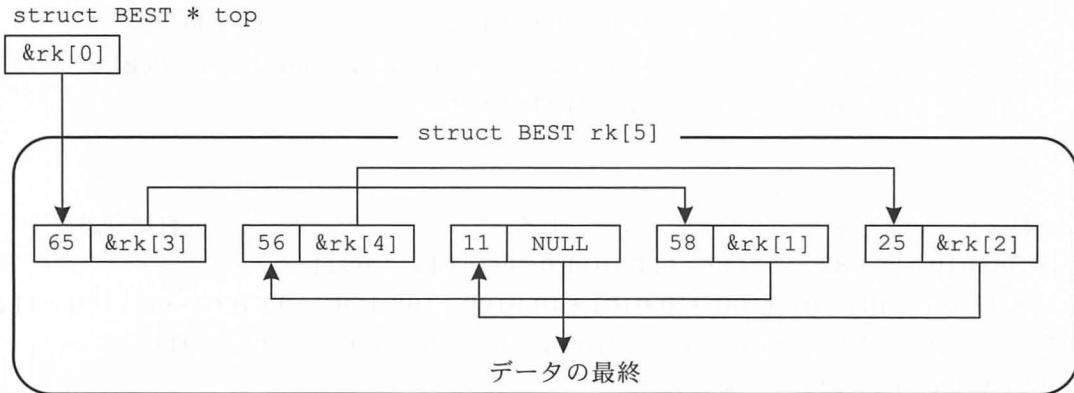
```

47             op = op->next;
48         }
49         op->next = [③];
50     }
51     else {
52         top = last;
53     }
54     last->next = [④];
55
56     op = ins;
57     while (op->next != last) {
58         op = op->next;
59     }
60     op->next = [⑤];
61 }
62
63 /* 表示部 */
64 printf("rank score\n");
65 op = top;
66 i = 1;
67 while (op != NULL) {
68     printf("%4d:%5d\n", i, op->score);
69     op = op->next;
70     i++;
71 }
72
73 /* データ入力部 */
74 printf("Input Score:");
75 scanf("%d", &iscore);
76 }
77
78 return 0;
79 }

```

ヒント

自分自身と同じ型の構造体を指すポインタを含む構造体の配列により、リスト構造を表現している。



1. 数の表現と処理

1 1ビットで2¹個、2ビットで2²個、3ビットで2³個の組合せがある。

したがって2ⁿ≥26の条件を満たす整数nを求めればよい。

条件より2ⁿ≥26、また2⁴=16、2⁵=32である。

nは整数であるのでn=5となる。 答 5ビット

2 ①64 ②32 ③6 ④6 ⑤6

3 $10^{n-1} < 16^{12} - 1 < 10^n$

$$\log_{10} 10^{n-1} < \log_{10} (16^{12} - 1) < \log_{10} 10^n$$

$$n-1 < \log_{10} 16^{12} - \log_{10} 1 < n$$

$$n-1 < 12 \log_{10} 16 - 0 < n$$

$$n-1 < 12 \log_{10} 2^4 < n$$

$$n-1 < 12 \times 4 \log_{10} 2 < n$$

$$n-1 < 14.448 < n$$

答 15けた

4 $D = \log_{10} 2^B = B \log_{10} 2$ 答 工

5 (1) ①1010 0101 ②A5 (2) ①1 0000 1110 ②10E (3) ①101 0000 ②50

(4) ①10 1011. 01 ②2B.4 (5) ①11. 0101 11 ②3.5C (6) ①111 0110. 11 ②76.C

(7) ①101 1100 ②5C ③92

6 ①1111 0011. 0001 ②108.25 ③10 0101 0000 ④356.8 ⑤1110 1101

7 ① $\frac{101}{128}$ ② $\frac{41}{64}$ ③ $\frac{23}{32}$

④B.18 ⑤DA.8 ⑥2.24 ⑦0. 1001 1 ⑧0. 1011 1001 ⑨0. 0111 ⑩0. 0010 01

8 ①2591.25 ②1010 0001 1111. 01 ③287.D

9 $(5 \times N^2 + 1 \times N + 4) \div 5 = (1 \times N^2 + 0 \times N + 2)$

$(5 \times N^2 + 1 \times N + 4) = (1 \times N^2 + 0 \times N + 2) \times 5$

$$5 \times N^2 + 1 \times N + 4 = 5 \times N^2 + 10$$

N=6 答 6進法

10 $(1 \times 2^{-2} + 1 \times 2^{-4})_2 \div (1 \times 2^{-3})_2 = 1 \times 2^1 + 1 \times 2^{-1} = (10.1)_2$ 答(10.1)₂

11 $0.05 \times 2 = 0.1$ $0.1 \times 2 = 0.2$ $0.2 \times 2 = \underline{0.4}$ $0.4 \times 2 = 0.8$ $0.8 \times 2 = 1.6$ $0.6 \times 2 = 1.2$ $0.2 \times 2 = \underline{0.4}$

ここから無限に繰り返す 答 0.05

12 ① $\frac{1}{16}$ ② $\frac{11}{16}$ ③ $\frac{1}{256}$ ④ $\frac{255}{256}$ ⑤ $\frac{168}{256}$ $\left(\frac{21}{32}\right)$

13 ①0.0001 ②0.1111 1111 ③0.1110 1 ④10.0010 01

14 ①454.5 ②258 ③279 ④997.125 ⑤D5A0 ⑥1110 0111.1

15 ①161.25 ②15 ③694 ④0011 0110 0101 ⑤757.125 ⑥0111 0101 0011

16 ①0.0001 1001 ②0.1 ③0.1011 0011 ④0.1010 0011 ⑤0.1101 1100

17 ①254 ②1010 0011 0101 1101 ③1110 1101 0110 ④141 266 ⑤C5 ⑥BE2

18 ①-2326 ②10111000 ③1110 0010 ④1111 0100

19 -2357

20 8000

21 ①情報落ち ②けた落ち ③オーバフロー ④アンダフロー ⑤丸め誤差 ⑥打切り誤差

22 ①1010 1010 ②-82 ③1111 1111 1011 0100 ④1111 0011

23 ①37-28→10 0101-01 1100→10 0101+10 0100=100 1001→00 1001→001 001=(11)₈

②4A-37→0100 1010-0011 0111→0100 1010+1100 1001=1 0001 0011

→0001 0011→010 011→(23)₈

24 ①DD ②92 ③FF ④80

25 -32768

練習問題 解答

26 ①101 1001. 1010 0011 ②10 0011. 1011 0011

27 ①イ ②エ ③キ ④ツ ⑤ケ

28 $(0.375)_{10} = (0.011)_2 = (0.11)_2 \times 2^{-1}$

指数部-1を7ビットで2の補数表示すると $(111\ 1111)_2$ となる。

答 0111 1111 1100 0000 0000 0000 0000 0000

29 ABCD, またはEBCD, または6BCD

30 ①BCCA ②48 330 ③-17 206 ④3 020.625 ⑤ $-(0.\ CA)_{16} \times 16^{-4}$

31 1000 0011 0100 1101

32 1100 0110

33 ①オ ②エ ③ウ ④カ ⑤ア ⑥キ ⑦イ

34 ①1100 0010 ②1100 1001 ③1101 0100 ④1111 1001

35 0 1 3 4 5 符号

0000	0001	0011	0100	0101	1101
------	------	------	------	------	------

36 4 6 3

0011	0100	0011	0110	1100	0011
------	------	------	------	------	------

数値 数値 符号

37 ①44 ②C6 ③55 ④33 ⑤B8

38 ①16進数F8と排他的論理和をとる。 ②16進数F8と論理積をとる。

③16進数F8と論理和をとる。 ④16進数F8と否定論理積をとる。

39 エ

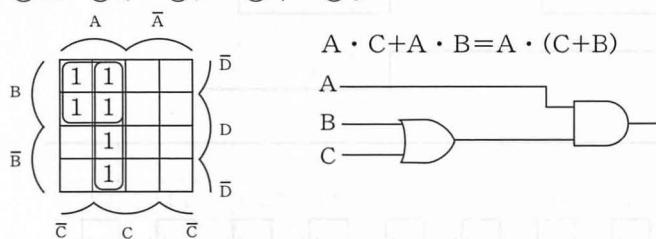
40 イ

41 7F800000

2. コンピュータの基本回路

1 ①エ ②オ ③ア ④イ ⑤ウ

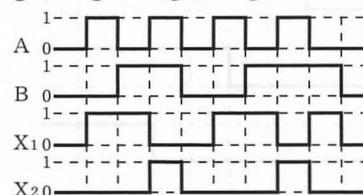
2



3 ①オ ②カ ③ア ④エ

4 ①イ ②イ ③イ ④ア

5



6 1011のとき

7 ① 入力 出力 ②ウ ③イ ④ア ⑤キ

A	B	X ₁	X ₂	X ₃
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

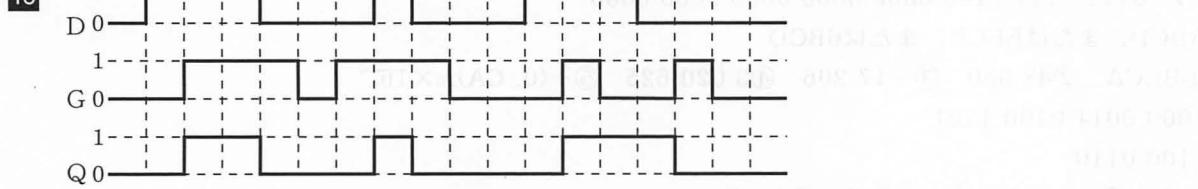
練習問題 解答

8 ①X₁ ②X₆ ③G₃ ④6

	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀
	1	0	1	0	1	1	0	0

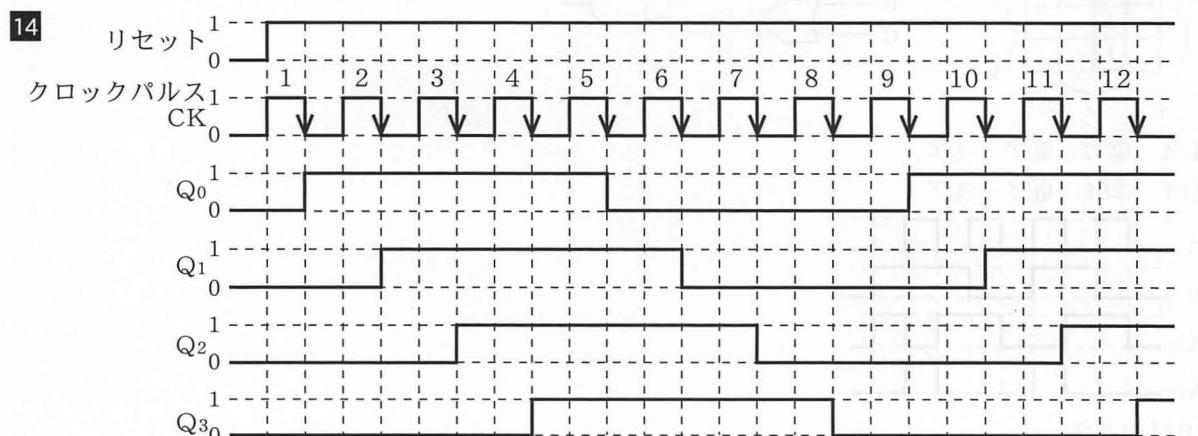
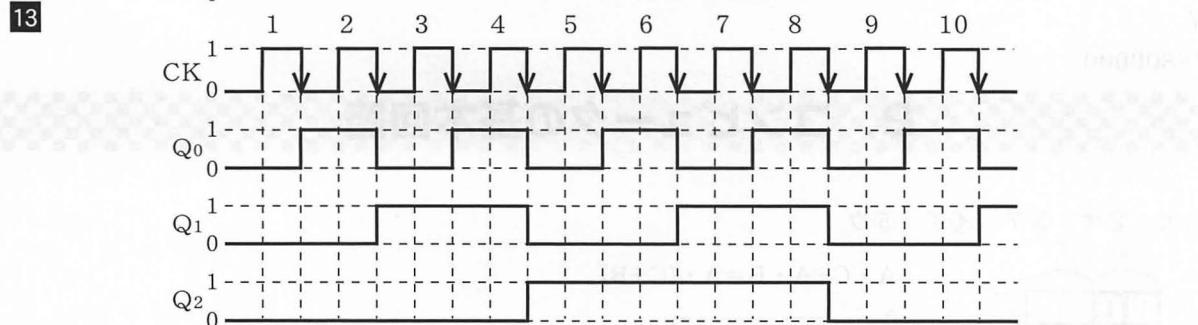
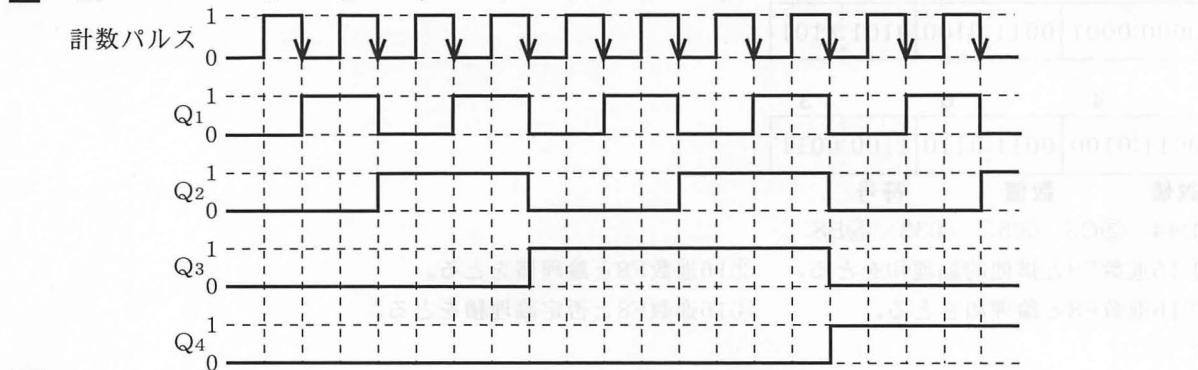
9

D 0
G 0
Q 0



11 ①5MHz ②Q₀:1 Q₁:0 Q₂:1 Q₃:0

12 (1) 計数パルス (2)1010



15 ①0.625MHz ②Q₀:0 Q₁:1 Q₂:1 Q₃:0 ③Q₁とQ₃

16 ①9 ②22 ③4Hz

練習問題 解答

17	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
	1	0	1	1	1	1	1	1

18 ①0 ②0 ③0 ④1 ⑤1

19 スイッチのチャタリングの影響をとる。Q : 1, \bar{Q} : 0。点灯する。

20 (1)①ウ ②ア ③エ ④イ (2)0.02V

21 ①2 ②A

22 ①イ ②ウ ③ア

3. コンピュータの基本構成と各部の働き

1 ①キ ②ク ③ケ ④ア ⑤カ ⑥エ ⑦コ ⑧ウ ⑨オ ⑩イ

2 ①ク ②ウ ③カ ④キ ⑤オ

3 ①システムバス ②周辺LSI ③チップセット ④ポート

4 76

5 ①ケ ②カ ③ク ④キ ⑤サ ⑥オ ⑦エ ⑧ア ⑨ウ ⑩イ

6 ①ウ ②ア ③エ ④イ ⑤オ

7 ①ア ②ウ ③オ ④イ ⑤エ

8 ①ウ ②イ

9 1命令の平均命令実行時間= $1.0 \times 0.6 + 4.5 \times 0.2 + 5.0 \times 0.2 = 2.5(\mu\text{秒})$

1秒間の命令実行数= $1 \div (2.5 \times 10^{-6}) = 0.4 \times 10^6$ 答 0.4MIPS

10 $(15 \times 10^6 \times 0.8) \div (60 \times 10^4) = 0.2 \times 10^2 = 20$ 答 20(件／秒)

11 $240 \div 10 = 24$ $24 - 5 = 19$ $19 + 1 = 20$ 答 20命令

12 $300 + 10 = 310$ 答 310

13 ①SIMD(Single Instruction / Multiple Data) ②MISD(Multiple Instruction / Single Data)

③MIMD(Multiple Instruction/Multiple Data) ④SISD(Single Instruction / Single Data)

14 ①カ ②キ ③サ ④エ ⑤オ ⑥イ ⑦ス ⑧ソ ⑨コ ⑩ク

15 ①フラッシュメモリ ②SRAM ③DRAM ④マスクROM

16 実行アクセス時間=キャッシュメモリアクセス時間×キャッシュメモリヒット率

+ 主記憶アクセス時間×(1-キャッシュメモリヒット率)

実行アクセス時間= $10 \times 0.7 + 50 \times (1 - 0.7) = 22$ 答 22n秒

17 ①ライトスルー ②メモリインタリーブ ③ライトバック

18 ①LRU(Least Recently Used) ②FIFO(First In First Out)

③LIFO>Last In First Out) ④LFU(Least Frequently Used)

19 ①ク ②キ ③ケ ④ア ⑤カ ⑥エ ⑦コ ⑧ウ ⑨オ ⑩イ

20 ①ア ②イ ③イ ④ア ⑤イ ⑥ア ⑦イ ⑧ア ⑨エ ⑩ウ

21 ①ソ ②ケ ③サ ④ア ⑤ウ ⑥ス ⑦カ ⑧セ

22 ①キ ②エ ③ア ④カ ⑤イ

23 ①コ ②ク ③ウ ④カ ⑤キ

24 $512 \times 9 \times 160 = 737\,280$ $737\,280 \div 1024 = 720$ 答 720キロバイト

25 $256 \times 10 \times 20480 = 52\,428\,800$ $52\,428\,800 \div 1024 \div 1024 = 50$ 答 50ギガバイト

26 ウ

27 ①10 ②5 ③4 ④16

28 ウ

29 イ

30 ①パケットライティング ②ディスクアットワنس ③トラックアットワنس

31 100→130→150→200→90→80→70→50の順に移動する。 答 250

練習問題 解答

- 32 ①ウ ②オ ③工 ④ア ⑤オ
33 ①イ ②エ ③ウ
34 $10\text{cm} \times 5\text{cm} = 4\text{インチ} \times 2\text{インチ} = 600 \times 4 \times 600 \times 2 = 2,880,000$ ドット
24ビット=3バイト $3 \times 2,880,000 = 8,640,000$ 8640000バイト≈8.6Mバイト
答 8.6Mバイト
35 $1 \times 720 \times 480 \times 30 \times 60 = 622,080,000$ 答 約622Mバイト
36 ①磁気カード ②ICカード ③RFID ④QRコード
37 ① $800 \times 600 \times 3$ バイト(24ビット)= $1,440,000$ バイト≈1.4Mバイト 答 1.4Mバイト
② $2^{24} = 16,777,216 \approx 1678$ 答 1678万色
③ $1.44\text{Mバイト} = 1,440,000\text{バイト}$, $1,440,000\text{バイト} \div (2\text{バイト} \times 22,000) \approx 32.7$
33秒は記録できないので、切り捨てて32秒 答 32秒
38 ①工 ②ア ③ア
39 ①ク ②オ ③キ ④カ ⑤ア

4. 通信

- 1 (1)24ビット=3バイト $3 \times 1,200 \times 1,000 = 3,600,000$ 3 600 000バイト=3 600kバイト
答 3 600kバイト
(2) $3,600 \div 20 = 180$ 14.4Mビット=1.8Mバイト=1 800kバイト
 $1,800 \times 0.5 = 900$ $900 \div 180 = 5$ 答 5枚
2 (1)1文字あたり必要なデータは $1+8+1=10$ ビット $300 \div 10 = 30$ 答 30文字／秒
(2) $(350 \times 1,024) \div (56,000 \times 0.512 \div 8) = 100$ 答 100秒
3 (1)4位相変調の場合1回の変調で2ビット伝送できる。 $2,400 \times 2 = 4,800$ 答 4 800bps
(2) $(96 \times 1,000) \div (4,800 \times 0.8 \div 8) = 200$ 答 200秒
4 20秒
5 ①ケ ②オ ③カ ④キ ⑤イ ⑥ア
6 $512 \div (64 \div 8 \times 0.5) = 128$ 答 128秒
7 ①工 ②イ ③キ ④カ ⑤オ
8 64kbps=8kバイト／秒 $2 \times 1.2 = 2.4$ kバイト $2.4 \div 8 = 0.3$ 秒
 $0.3 \div 2 = 0.15$ 答 工
9 $(768\text{バイト} \times 5 + 256) \times 8 = 32,768\text{ビット} = 32\text{kビット}/5\text{件}$
 $18,000 \div 3,600 = 5\text{件}/\text{秒}$ $64 \div 32 = 0.5$ 答 50%
10 ①ウ ②イ ③工 ④ア
11 ①キ ②ケ ③コ ④ア ⑤ウ ⑥サ
12 ①ウ ②カ ③ア ④ク ⑤工
13 ①エ ②ク ③カ ④イ ⑤セ ⑥チ

5. ソフトウェアの基礎

- 1 ①キ ②オ ③工 ④イ ⑤カ
2 ①ク ②ケ ③カ ④エ ⑤キ
3 ①エ ②ア ③ウ ④カ ⑤オ
4 ①ウ ②シ ③サ ④オ ⑤ソ ⑥ケ ⑦エ ⑧カ ⑨ク ⑩ア
5 ①ウ ②エ ③イ ④カ ⑤ケ
6 ①ア ②キ ③イ ④ク ⑤ウ
7 ①ソ ②ス ③シ ④エ ⑤ア ⑥ク ⑦ウ ⑧イ ⑨セ ⑩カ
8 ①ケ ②キ ③カ ④オ ⑤ク ⑥シ ⑦ス ⑧エ ⑨ウ ⑩イ

- 9 ①イ ②カ ③ウ ④ア ⑤キ
 10 ①エ ②カ ③サ ④ク ⑤イ ⑥オ ⑦ウ ⑧セ ⑨ア ⑩ス ⑪コ
 11 ①オ ②エ ③ク ④ア ⑤カ
 12 ①カ ②イ ③ア ④ケ ⑤エ
 13 ①カ ②オ ③ウ ④キ ⑤エ
 14 ①コ ②ク ③サ ④キ ⑤ケ ⑥カ ⑦ウ ⑧オ ⑨ス ⑩シ (⑦, ⑧, ⑨は順不同)
 15 ①ク ②カ ③キ ④ケ ⑤オ ⑥コ ⑦シ ⑧ア ⑨エ ⑩イ
 16 ①サ ②エ ③ケ ④カ ⑤ア ⑥ク ⑦ウ ⑧キ ⑨シ ⑩コ

6. その他の情報関連知識

- 1 ①ケ ②ウ ③ア ④サ ⑤コ ⑥オ ⑦ク ⑧カ ⑨キ ⑩エ
 2 ①エ ②ウ ③ク ④イ ⑤コ
 3 ①コ ②ク ③エ ④カ ⑤ケ
 4 ①ウ ②カ ③エ ④キ ⑤コ
 5 ①ケ ②イ ③ウ ④ク ⑤オ ⑥ス ⑦カ ⑧ソ ⑨エ ⑩コ
 6 (1)イ (2)ア, ウ, エ
 7 ①シ ②ク ③エ ④ウ ⑤ス ⑥オ ⑦ア ⑧キ
 8 ①オ ②ク ③カ ④ウ ⑤ア ⑥キ
 9 ①エ ②オ ③ア ④ウ ⑤イ
 10 ①オ ②シ ③キ ④ク ⑤ア ⑥サ ⑦カ ⑧ウ ⑨コ ⑩ケ
 11 (1) $0.9 \times 0.9 = 0.81$
 (2) $1 - (1 - 0.9) \times (1 - 0.9) = 1 - 0.1 \times 0.1 = 1 - 0.01 = 0.99$
 (3) $\{1 - (1 - 0.9) \times (1 - 0.9)\} \times 0.9 = 0.99 \times 0.9 = 0.891$

12 点検を行わない場合 $\frac{1980}{1980+20} = 0.99$

点検を行う場合 $\frac{995}{995+5} = 0.995$

- 13 ①エ ②イ ③ウ ④カ
 14 ①イ ②エ ③ウ ④カ ⑤オ
 15 ①コ ②ウ ③カ ④イ ⑤ク
 16 ①ア ②エ ③ケ ④オ ⑤イ ⑥ク ⑦カ ⑧コ ⑨キ ⑩ウ
 17 ①シ ②ク ③オ ④エ ⑤キ ⑥イ ⑦ア ⑧ウ ⑨ソ ⑩サ
 18 ①オ ②ケ ③エ ④ウ ⑤ク ⑥ア ⑦イ ⑧カ
 19 ①エ ②ウ ③カ ④ア ⑤イ
 20 ①ク ②ウ ③イ ④ケ ⑤オ ⑥セ ⑦ア ⑧コ ⑨エ ⑩ス
 21 ①カ ②ウ ③ケ ④エ ⑤ア

7. アルゴリズム

- 1 ①n-1 ②i+1 ③d(j-1) ④d(j)
 2 ①n-1 ②i ③j-1 ④d(j+1)
 3 ①1 ②d(i) ③i+1 ④> ⑤≤
 4 ①d(n+1) ②key ③i+1 ④> ⑤≤
 5 ①イ ②オ ③ウ ④キ
 6 ①0 ②1 ③2 ④SUM+L→SUM ⑤L→J
 7 ①1→F ②n:1 ③2×F+1→F
 8 ①10 ②M(K) ③N(K)+D ④M(K)×D

- 9 ①i ②SUM+j → SUM ③= ④j ⑤j+1 → j
- 10 ①0 → C ②X²+Y²≤1 ③C+1 → C ④C÷N×4 → P
- 11 ①イ ②シ ③ウ ④ク ⑤コ
- 12 ①カ ②エ ③キ ④ア ⑤イ
- 13 ①ウ ②ア ③キ ④オ ⑤ケ
- 14 ①エ ②ウ ③オ ④オ ⑤イ
- 15 ①1÷n ②x+dx ③(y₁+y₂)×dx÷2 ④S+ds ⑤S×4
- 16 ①h÷n ②n ③(n-i)÷n×r ④V+Vi
- 17 ①ア ②カ ③オ ④キ ⑤エ
- 18 ①ST ②10 ③VAL×VAL ④VAL (ST+Dでも可) ⑤KETA+1

8. CASL II プログラムの作成能力

- 1 (1) ①SLA GR1, 1 ②LD GR2, GR1 ③SLA GR1, 2
④ADDA GR1, GR2 (2) 実行後のGR1の値 0000 0000 0000 0010
- 2 ①GR4, 1 ②GR1, 1, GR1 ③S2 ④S3
- 3 ①GR3, LTBL, GR3 ②GR0, GR2 ③JPL ④JMI
- 4 ①GR4, 2, GR1 ②GR1, 0, GR4 ③GR1, 0 ④GR1, 1, GR2
- 5 ①SRET ②0, GR1 ③FFFF ④GR1, 1, GR1
- 6 ①0, GR3 ②GR3, GR0 ③GR3, 0, GR2 ④GR0, 0, GR2
- 7 ①GR4, GR0 ②GR3, =1 ③GR1, 1 ④GR0, =#8000
- 8 ①P1 ②P2 ③FIN ④FIN
- 9 (1) ①GR4, GR3またはGR4, =0 ②LBL1 ③GR2, -1, GR2またはGR4, 1, GR4 (2) 20回
⑩ ①JZE ②JMI ③JPL ④GR1, 1, GR1 ⑤ADDAまたはADDL
- 11 ①GR4, GR0 ②GR3, 0, GR2 ③GR2, 2, GR1 ④GR0, GR3
- 12 ①GR3, GR4 ②GR0, GR5 ③GR3, GR0 ④GR2, SAVE
- 13 ①GR4, 0, GR3 ②GR1, 0, GR3 ③LAST ④LOOP
- 14 ①0, GR2 ②GR0 ③GR3 ④0, GR3
- 15 ①GR3, LENG ②BEGIN ③GR1, 0 ④GR0, INBUF, GR1
⑤GR2, INBUF, GR1
- 16 ①GR4, 1, GR1 ②GR3, GR3 ③GR3, 0 ④JMI ⑤GR2, 0, GR1
- 17 ①GR1, GR1 ②GR1, 1 ③GR0, OBUF, GR2 ④JMI ⑤GR2, OLNG
- 18 ①OUTPUT ②RANK ③GR3, 3 ④GR2 ⑤18
- 19 ①GR0, 0 ②NUM ③AND ④OR ⑤LOOP
- 20 ①GR2, 0 ②XOR ③GR1, GR1 ④GR2, 1, GR2 ⑤SLL
- 21 ①NOTMIN ②NOTMAX ③LOOP ④SUBA ⑤SUBA
- 22 ①51, GR1 ②ST ③1, GR3 ④0, GR4 ⑤0, GR2
- 23 ①GR3, DATA ②D30 ③D20 ④D10 ⑤D40
- 24 ①JPL ②GR0, 1 ③GR0, 2 ④GR1, 1, GR1 ⑤JZE
- 25 ①DRET ②SLL ③SRL ④GR3, SHO ⑤GR1, GR2
- 26 ①ARTN ②GR4, 1, GR4 ③GR3, -1, GR3 ④GR5, GR2
⑤GR5, -1, GR5
- 27 ①LD ②BGN ③FIN ④CLS ⑤EXTBGN

9. Cプログラムの作成能力

- 1 ①> ②== ③% ④dvi

練習問題 解答

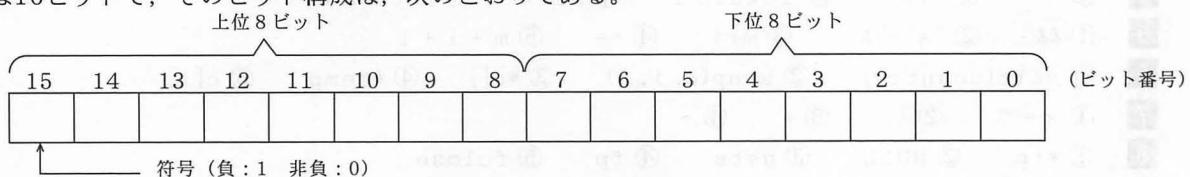
- 2 ① *a != -99 ② a++ ③ int *a ④ wa + *a ⑤ a++
3 ① mojis(u(moji)) ② != ③ != ④ && ⑤ 'a'-'A'
4 ① == ② '¥0' ③ return 1 ④ m1++ ⑤ return 0
5 ① && ② 'a'-'A' ③ m++ ④ == ⑤ m + i + 1
6 ① sort(country) ② swap(c, i, j) ③ *c[] ④ *temp ⑤ c[i]
7 ① x = 0 ② || ③ + ④ -
8 ① *fp ② NULL ③ gets ④ fp ⑤ fclose
9 ① fopen ② == ③ fp ④ SEEK_SET ⑤ fscanf
10 ① mid ② lp++ ③ rp-- ④ rp - 1 ⑤ rp + 1
11 ① 1 ② ch ③ ch + 1 ④ ch ⑤ 2 * pa
12 ① || ② 0 ③ 0 ④ !=
13 ① break ② x ③ MAX ④ 1.0 / s
14 ① (a + b) ② b = c ③ a = c ④ x * x * x - q
15 ① q ② fx ③ fd ④ cnt++
16 ① break ② continue ③ menseki ④ h
17 ① == ② < ③ n * 2 ④ +=
18 ① &a ② b ③ i-- ④ b++
19 ① (n + 1) ② n * n ③ n ④ 1
20 ① != ② == ③ != ④ == ⑤ !=
21 ① op != NULL ② iscore ③ last ④ ins ⑤ NULL

アセンブラー言語の仕様 (独立行政法人・情報処理推進機構)

1. システムCOME IIの仕様

1. 1 ハードウェアの仕様

(1) 1語は16ビットで、そのビット構成は、次のとおりである。



(2) 主記憶の容量は65536語で、そのアドレスは0～65535番地である。

(3) 数値は、16ビットの2進数で表現する。負数は、2の補数で表現する。

(4) 制御方式は逐次制御で、命令語は1語長又は2語長である。

(5) レジスタとして、GR (16ビット) , SP (16ビット) , PR (16ビット) , FR (3ビット) の4種類がある。

GR (汎用レジスタ, General Register) は、GR0～GR7の8個があり、算術、論理、比較、シフトなどの演算に用いる。このうち、GR1～GR7のレジスタは、指標レジスタ (index register) としてアドレスの修飾にも用いる。

SP (スタックポインタ, Stack Pointer) は、スタックの最上段のアドレスを保持している。

PR (プログラムレジスタ, Program Register) は、次に実行すべき命令語の先頭アドレスを保持している。

FR (フラグレジスタ, Flag Register) は、OF (Overflow Flag) , SF (Sign Flag) , ZF (Zero Flag) と呼ぶ3個のビットからなり、演算命令などの実行によって次の値が設定される。これらの値は、条件付き分岐命令で参照される。

OF	算術演算命令の場合は、演算結果が-32768～32767に収まらなくなったとき1になり、それ以外のとき0になる。論理演算命令の場合は、演算結果が0～65535に収まらなくなったとき1になり、それ以外のとき0になる。
SF	演算結果符号が負 (ビット番号15が1) のとき1、それ以外のとき0になる。
ZF	演算結果符号が零 (全部のビットが0) のとき1、それ以外のとき0になる。

(6) 論理加算又は論理減算は、被演算データを符号のない数値とみなして、加算又は減算する。

1. 2 命令

命令の形式及びその機能を示す。ここで、一つの命令コードに対し2種類のオペランドがある場合、上段はレジスタ間の命令、下段はレジスタと主記憶間の命令を表す。

命 令	書 き 方		命 令 の 説 明	FRの設定
	命 令 コ ー ド	オ ペ ラ ン ド		

(1)

ロード LoaD	LD	r1, r2 r, adr[, x]	r1 ← (r2) r1 ← (実効アドレス)	○*1
ストア StOre	ST	r, adr[, x]	実効アドレス ← (r)	-
ロードアドレス Load Address	LAD	r, adr[, x]	r ← 実効アドレス	

(2) 算術、論理演算命令

算術加算 ADD Arithmetic	ADDA r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) + (r_2)$ $r \leftarrow (r) + (\text{実効アドレス})$	○
論理加算 ADD Logical	ADDL r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) + L(r_2)$ $r \leftarrow (r) + L(\text{実効アドレス})$	
算術減算 SUBtract Arithmetic	SUBA r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) - (r_2)$ $r \leftarrow (r) - (\text{実効アドレス})$	○
論理減算 SUBtract Logical	SUBL r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) - L(r_2)$ $r \leftarrow (r) - L(\text{実効アドレス})$	
論理積 AND	AND r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) \text{ AND } (r_2)$ $r \leftarrow (r) \text{ AND } (\text{実効アドレス})$	○*1
論理和 OR	OR r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) \text{ OR } (r_2)$ $r \leftarrow (r) \text{ OR } (\text{実効アドレス})$	
排他的論理和 eXclusive OR	XOR r_1, r_2 $r, \text{adr}[, x]$	$r_1 \leftarrow (r_1) \text{ XOR } (r_2)$ $r \leftarrow (r) \text{ XOR } (\text{実効アドレス})$	

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	r_1, r_2 $r, \text{adr}[, x]$	(r_1) と (r_2) , 又は (r) と (実効アドレス) の算術比較又は論理比較を行い, 比較結果によって, FRに次の値を設定する。	○*1																								
論理比較 ComPare Logical	CPL	r_1, r_2 $r, \text{adr}[, x]$	<table border="1"> <thead> <tr> <th>比較結果</th> <th colspan="2">FRの値</th> </tr> <tr> <th></th> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>$(r_1) > (r_2)$</td> <td>0</td> <td>0</td> </tr> <tr> <td>$(r) > (\text{実効アドレス})$</td> <td>0</td> <td>1</td> </tr> <tr> <td>$(r_1) = (r_2)$</td> <td>1</td> <td>0</td> </tr> <tr> <td>$(r) = (\text{実効アドレス})$</td> <td>1</td> <td>0</td> </tr> <tr> <td>$(r_1) < (r_2)$</td> <td>0</td> <td>1</td> </tr> <tr> <td>$(r) < (\text{実効アドレス})$</td> <td>0</td> <td>1</td> </tr> </tbody> </table>		比較結果	FRの値			SF	ZF	$(r_1) > (r_2)$	0	0	$(r) > (\text{実効アドレス})$	0	1	$(r_1) = (r_2)$	1	0	$(r) = (\text{実効アドレス})$	1	0	$(r_1) < (r_2)$	0	1	$(r) < (\text{実効アドレス})$	0	1
比較結果	FRの値																											
	SF	ZF																										
$(r_1) > (r_2)$	0	0																										
$(r) > (\text{実効アドレス})$	0	1																										
$(r_1) = (r_2)$	1	0																										
$(r) = (\text{実効アドレス})$	1	0																										
$(r_1) < (r_2)$	0	1																										
$(r) < (\text{実効アドレス})$	0	1																										

(4) シフト演算命令

算術左シフト Shift Left Arithmetic	SLA $r, \text{adr}[, x]$	符号を除き (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果, 空いたビット位置には, 左シフトのときは0, 右シフトのときは符号と同じものが入る。	○*2
算術右シフト Shift Right Arithmetic	SRA $r, \text{adr}[, x]$		
論理左シフト Shift Left Logical	SLL $r, \text{adr}[, x]$	符号を含み (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果, 空いたビット位置には0が入る。	
論理右シフト Shift Right Logical	SRL $r, \text{adr}[, x]$		

(5) 分岐命令

正分岐 Jump on PLus	JPL $\text{adr}[, x]$	FRの値によって, 実効アドレスに分岐する。分岐しないときは, 次の命令に進む。	<table border="1"> <thead> <tr> <th rowspan="2">命令</th> <th colspan="3">分岐するときのFRの値</th> </tr> <tr> <th>OF</th> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>JPL</td> <td></td> <td>0</td> <td>0</td> </tr> <tr> <td>JMI</td> <td></td> <td>1</td> <td></td> </tr> <tr> <td>JNZ</td> <td></td> <td></td> <td>0</td> </tr> <tr> <td>JZE</td> <td></td> <td></td> <td>1</td> </tr> <tr> <td>JOV</td> <td>1</td> <td></td> <td></td> </tr> </tbody> </table>	命令	分岐するときのFRの値			OF	SF	ZF	JPL		0	0	JMI		1		JNZ			0	JZE			1	JOV	1		
命令	分岐するときのFRの値																													
	OF	SF	ZF																											
JPL		0	0																											
JMI		1																												
JNZ			0																											
JZE			1																											
JOV	1																													
負分岐 Jump on MIminus	JMI $\text{adr}[, x]$																													
非零分岐 Jump on Non Zero	JNZ $\text{adr}[, x]$																													
零分岐 Jump on ZERO	JZE $\text{adr}[, x]$																													
オーバーフロー分岐 Jump on OVerflow	JOV $\text{adr}[, x]$																													
無条件分岐 unconditional JUMP	JUMP $\text{adr}[, x]$																													
		無条件に実効アドレスに分岐する。																												

(6) スタック操作命令

プッシュ PUSH	PUSH adr [, x]	$SP \leftarrow (SP) - L 1,$ $(SP) \leftarrow$ 実効アドレス	-
ポップ POP	POP r	$r \leftarrow ((SP)),$ $SP \leftarrow (SP) + L 1$	-

(7) コール, リターン命令

コール CALL subroutine	CALL adr [, x]	$SP \leftarrow (SP) - L 1,$ $(SP) \leftarrow (PR),$ $PR \leftarrow$ 実効アドレス	-
リターン Return from subroutine	RET	$PR \leftarrow ((SP)),$ $SP \leftarrow (SP) + L 1$	-

(8) その他

スーパバイザコール SuperVisor Call	SVC adr [, x]	実効アドレスを引数として割出しを行う。 実行後のGRとFRは不定となる。	-
ノーオペレーション No OPeration	NOP	何もしない。	-

- (注) r, r1, r2 いずれもGRを示す。指定できるGRはGR0～GR7
 adr アドレスを示す。指定できる値の範囲は0～65535
 x 指標レジスタとして用いるGRを示す。指定できるGRはGR1～GR7
 [] []内の指定は省略できることを示す。
 () ()内のレジスタはアドレスに格納されている内容を示す。
 実効アドレス adrとxの内容との論理加算値又はその値が示す番地
 ← 演算結果を、左辺のレジスタ又はアドレスに格納することを示す
 +L, -L 論理加算、論理減算を示す。
 FRの設定 ○ : 設定されることを示す。
 ○*1 : 設定されることを示す。ただし、OFには0が設定される。
 ○*2 : 設定されることを示す。ただし、OFにはレジスタから最後に送り出されたビットの値が設定される。
 - : 実行前の値が保持されることを示す。

1.3 文字の符号表

- (1) JIS X 0201 ラテン文字・片仮名用8ビット符号で規定する文字の符号表を使用する。
- (2) 右に符号表の一部を示す。1文字は8ビットからなり、上位4位4ビットを列で、下位4ビットを行で示す。例えば、間隔、4, H, ¥のビット構成は、16進表示で、それぞれ20, 34, 48, 5Cである。16進表示で、ビット構成が21～7E（及び表では省略しているA1～DF）に対応する文字を図形文字という。図形文字は表示（印刷）装置で、文字として表示（印刷）できる。
- (3) この表にない文字とそのビット構成が必要な場合は、問題中で与える。

行 \ 列	02	03	04	05	06	07
0	間隔	0	@	P	'	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	,	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[k	{
12	,	<	L	¥	l	
13	-	=	M]	m	}
14	.	>	N	^	n	~
15	/	?	O	_	o	

2. アセンブラー言語CASL IIの仕様

2. 1 言語の仕様

- (1) CASL IIは、COMET IIのためのアセンブラー言語である。
- (2) プログラムは、命令行及び注釈行からなる。
- (3) 1命令は1命令行で記述し、次の行へ継続できない。
- (4) 命令行及び注釈は、次に示す記述の形式で、行の1文字目から記述する。

行の種類		記述の形式
命令行	オペランドあり	[ラベル] {空白} {命令コード} {空白} {オペランド} [{空白}[コメント]]
	オペランドなし	[ラベル] {空白} {命令コード} [{空白}[{:} [コメント]]]
注釈行		[空白] {:} [コメント]

(注) []	[]内の指定が省略できることを示す。
{ }	{ }内の指定が必須であることを示す。
ラベル	その命令の（先頭の語の）アドレスを他の命令やプログラムから参照するための名前である。長さは1～8文字で、先頭の文字は英大文字でなければならない。以降の文字は、英大文字又は数字のいずれでもよい。なお、予約語であるGR0～GR7は、使用できない。
空白	1文字以上の間隔文字の列である。
命令コード	命令ごとに記述の形式が定義されている。
オペランド	命令ごとに記述の形式が定義されている。
コメント	覚え書きなどの任意の情報であり、処理系で許す任意の文字を書くことができる。

2. 2 命令の種類

命令は、4種類のアセンブラー命令 (START,END,DS,DC) , 4種類のマクロ命令 (IN,OUT,RPUSH,RPOP) 及び機械語命令 (COMET IIの命令) からなる。その仕様を次に示す。

命令の種類	ラベル	命令コード	オペランド	機能
アセンブラー命令	ラベル	START	[実行開始番地]	プログラムの先頭を定義 プログラムの実行開始番地を定義 他のプログラムを参照する入口名を定義
		END		プログラムの終わりを明示
	[ラベル]	DS	語数	領域を確保
	[ラベル]	DC	定数[, 定数]…	定数を定義
マクロ命令	[ラベル]	IN	入力領域, 入力文字長領域	入力装置から文字データを入力
	[ラベル]	OUT	出力領域, 出力文字長領域	出力装置へ文字データを出力
	[ラベル]	RPUSH		GRの内容をGRに格納
	[ラベル]	RPOP		スタックの内容をGRに格納
機械語命令	[ラベル]		(「1.2 命令」を参照)	

2. 3 アセンブラー命令

アセンブラー命令は、アセンブラーの制御などを行う。

- (1) START [実行開始番地]

START命令は、プログラムの先頭を定義する。

実行開始番地は、そのプログラム内で定義されたラベルで指定する。指定がある場合はその番地から、省略した場合はSTART命令の次の命令から、実行を開始する。

また、この命令につけられたラベルは、他のプログラムから入口名として参照できる。

(2)	END	
-----	-----	--

END命令は、プログラムの終わりを定義する。

(3)	DS	語数
-----	----	----

DS命令は、指定した語数の領域を確保する。

語数は、10進定数 (≥ 0) で指定する。語数を0とした場合、領域は確保しないが、ラベルは有効である。

(4)	DC	定数[, 定数] ...
-----	----	--------------

DC命令は、定数で指定したデータを（連続する）語に格納する。

定数には、10進定数、16進定数、文字定数、アドレス定数の4種類がある。

定数の種類	書き方	命令の説明
10進定数	n	nで指定した10進数値を、1語の2進数データとして格納する。ただし、nが-32768～32767の範囲にないときは、その下位16ビットを格納する。
16進定数	#h	hは4けたの16進数（16進数字は0～9, A～F）とする。hで指定した16進数値を1語の2進数データとして格納する（0000≤h≤FFFF）。
文字定数	'文字列'	文字列の文字数 (>0) 分の連続する領域を確保し、最初の文字は第1語の下位8ビットに、2番目の文字は第2語の下位8ビットに、…と順次文字でデータとして格納する。各語の上位8ビットには0のビットが入る。文字列には、間隔及び任意の図形文字を書くことができる。ただし、アポストロフィ(')は2個続けて書く。
アドレス定数	ラベル	ラベルに対応するアドレスを一語の2進数データとして格納する。

2.4 マクロ命令

マクロ命令は、あらかじめ定義された命令群とオペランドの情報によって、目的の機能を果たす命令群を生成する（語数は不定）。

(1)	IN	入力領域, 入力文字長領域
-----	----	---------------

IN命令は、あらかじめ割り当てられた入力装置から、1レコードの文字データを読み込む。

入力領域は、256語長の作業域のラベルであり、この領域の先頭から、1文字を1語に対応させて順次入力される。レコードの区切り符号（キーボード入力の復帰符号など）は格納しない。格納の形式は、DC命令の文字定数と同じである。入力データが256文字に満たない場合、入力領域の残りの部分は実行前のデータを保持する。入力データが256文字を超える場合、以降の文字は無視される。

入力文字長領域は、1語長の領域ラベルであり、入力された文字の長さ (≥ 0) が2進数で格納される。ファイルの終わり（end of file）を検出した場合は、-1が格納される。

IN命令を実行すると、GRの内容は保存されるが、FRの内容は不定となる。

(2)	OUT	出力領域, 出力文字長領域
-----	-----	---------------

OUT命令は、あらかじめ割り当てられた出力装置に、文字データを、1レコードとして書き出す。

出力領域は、出力しようとするデータが1文字1語で格納されている領域のラベルである。格納の形式は、DC命令の文字定数と同じであるが、上位8ビットは、OSが無視するので0でなくてもよい。

出力文字長領域は、1語長の領域ラベルであり、出力しようとする文字の長さ (≥ 0) を2進数で格納しておく。

OUT命令を実行するとGRの内容は保存されるが、FRの内容は不定となる。

(3)	RPUSH	
-----	-------	--

RPUSH命令は、GRの内容を、GR1, GR2, …, GR7の順序でスタックに格納する。

(4)	RPOP	
-----	------	--

RPOP命令は、スタックの内容を順次取り出し、GR7, GR6, …, GR1の順序でGRに格納する。

2.5 機械語命令

機械語命令のオペランドは、次の形式で記述する。

r, r1, r2	GRは、記号GR0～GR7で指定する。
x	指標レジスタとして用いるGRは、記号GR1～GR7で指定する。
adr	アドレスは、10進定数、16進定数、アドレス定数又はリテラルで指定する。
	リテラルは、一つの10進定数、16進定数又は文字定数の前に等号(=)を付けて記述する。CASL IIは、等号の後の定数をオペランドとするDC命令を生成し、そのアドレスをadrの値とする。

2.6 その他

- (1) アセンブラーによって生成される命令語や領域の相対位置は、アセンブラ言語での記述順序とする。ただし、リテラルから生成されるDC命令は、END命令の直前にまとめて配置される。
- (2) 生成された命令語、領域は、主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して、次の取り決めがある。

- (1) アセンブラーは、未定義ラベル（オペランド欄に記述されたラベルのうち、そのプログラム内で定義されていないラベル）を、他のプログラムの入口名（START命令のラベル）と解釈する。この場合、アセンブラーはアドレスの決定を保留し、その決定をOSに任せる。OSは、実行に先立って他のプログラムの入口名との連係処理を行いアドレスを決定する。（プログラムの連係）
- (2) プログラムは、OSによって起動される。プログラムがロードされる主記憶の領域は不定とするが、プログラムのラベルに対応するアドレス値が、OSによって実アドレスに補正されるものとする。
- (3) プログラムの起動時に、OSはプログラム用に十分な容量のスタック領域を確保し、その最後のアドレスに1加算した値をSPに設定する。
- (4) OSは、CALL命令でプログラムに制御を渡す。プログラムを終了しOSに制御を戻すときは、RET命令を使用する。
- (5) IN命令に対応する入力装置、OUT命令に対応する出力装置の割当ては、プログラムの実行に先立って利用者が行う。
- (6) OSは、入力装置や媒体による入出力手続きの違いを吸収し、システムでの標準の形式及び手続（異常処理を含む）で入出力を行う。したがって、IN,OUT命令では、入出力装置の違いを意識する必要はない。

3.2 未定義事項

プログラムの実行等に関し、この仕様で定義しない事項は、処理系によるものとする。

《索引》

英数字

2進カウンタ	25	D - F F	23	I Cカード	50
2分探索法	85	DMA	37	I D E	49
7セグメント	29	DMZ	74	I E E E - 1 3 9 4	49
8進カウンタ	27	DRAM	41	I M A P	72
ADDA	175	DS	178	I N	178
ADDL	175	DVD-R	46	I P	72
ADSL	57	DVD-W	46	I r D A	49
AMeDAS	70	DVD-ROM	46	I S D N	58
AND	11,175	EBCDICコード	10	I S Oコード	10
ASCIIコード	10	EEPROM	41	I T S	70
ASK	52	END	178	J a v a	62
ATM	71	EOF	141	J C L	61
AVI	78	EOS	70	J I Sコード	10
BASIC	62	E PROM	41	J K - F F	23
b a u d	52	EUCコード	10	J M I	175
BCDコード	5	EUCツール	65	J N Z	175
b i t / s	52	EWS	63	J O V	175
BMP	78	EX-OR	11	J P E G	78
b p s	52	F A (フルアダー)	18	J P L	175
C	62	F A(FactoryAutomation)	70	J U M P	175
C++	62	f c l o s e	141	J Z E	175
CAD	66	FD	46	L A D	174
CAI	70	F D D I	74	LAN	71
CALL	176	FF	23	LD	174
CASEツール	64	f g e t s	141	LFU	43
CASL II	106,177	FIFO	43	LIFO	43
CATV	57	FILE構造体	142	LISP	62
CD	71	FLOPS	38	LRU	43
CIM	70	f o p e n	141	MIDIファイル	78
CISC	38	Fortran	62	MIMD	40
COBOL	62	f p u t s	141	MIPS	38
COMET II	106,174	FSK	52	MISD	40
COSMETS	71	FTP	72	MO	46
CPA	175	FTTH	57	MOS型	42
CPI	39	GIF	78	MP3	78
CPL	175	GPIB	49	MPEG	78
CRT	48	GPS	71	NAND	11
CSMA/CD	74	GUI	66	NIC	74
CUI	66	HA	17	NNTP	72
DBMS	65	HDLC手順	53	NOP	176
DC	178	HDSL	57	NOT	11
DFD	69	HTML	72	NULL	141
		HUB	74	OA	71

O C R	48,67	S R L	175	位相変調方式	52		
O F	174	S T	174	一致回路	18		
OMR	48,67	S T A R T	177	イニシエータ	60		
O R	11,175	S T N	48	インクジェットプリンタ	48		
O S	60	s t r u c t	138	インストール	65		
O S I 参照モデル	72	S U B A	175	インターネット	72		
O U T	178	S U B L	175	インターフェース	31		
P C M 録音	78	S V C	176	インターリフタ	62		
P D P	48	T C P	72	インターネット	72		
P E R T	69	T C P / I P	72	インパクト型	48		
P N G	78	T e l n e t	72	ウォータフォールモデル	68		
P O P	176	T - F F	23	打ち切り誤差	7		
P O P 3	72	T F T	48	運用管理	60		
P O S	70	T I F F	78	エクストラネット	72		
P r o l o g	62	U n i c o d e	10	エグゼキューションサイクル	33		
P R O M	41	U S B	49	エミュレータ	62		
P S K	52	V A R	67	エラトステネスのふるい	96,155		
P U S H	176	V D S L	57	エンコーダ	29		
Q R コード	50	V D T	49	演算装置	33		
Q R コードリーダ	48	V D T 障害	80	エンドユーザ	63		
R A I D	44	W A N	71	応用ソフトウェア	65		
R A M	41	W A V ファイル	78	オーバーフロー	7		
R A S I S	60	W e b ページ	72	オーバーレイ	62		
R E T	176	W W W	72	オープンコレクタ	32		
R F I D	50	x D S L	57	オブジェクト指向	62		
R I S C	38	X O R	175	オブジェクトプログラム	62		
R O M	41	Z F	174	オペレーティングシステム	60		
R P O P	178	あ行					
R P U S H	178	アキュムレータ	33	か行			
R S - 2 3 2 C	49	アクセス権	74	カードリーダ	48		
R S - F F	23	アクセス時間	44	回線交換網	54		
S C S I	49	アセンブラー	62	階層モデル	67		
S D S L	57	アセンブラー言語	62	回転待ち時間	44		
S F	174	圧縮	78	外部設計	68		
S I M D	40	アップデート	66	カウンタ	24		
S I S	70	アップロード	72	仮数	7		
S I S D	40	アドレス	119,136	カスタマイズ	65		
S L A	175	アドレス演算子	136	仮想記憶管理	60		
S L L	175	アプリケーション層	72	仮想記憶方式	41		
S m a l l t a l k	62	アプリケーションソフトウェア	65	仮想コンピュータ	106		
S M T P	72	アルゴリズム	82	仮想端末	72		
S Q L	63	アロー演算子	139	カタログ機能	60		
S R A	175	暗号化	81	稼働率	75		
S R A M	41	アンダーフロ	7	カプセル化	64		

	さ 行		
可変区画方式	62	シリアル伝送方式	52
可用性	61	シリンド	44
カルノー図	13	シンセサイザ	79
簡易言語	67	伸張	78
関係モデル	67	振幅変調方式	52
換字式	81	シンプソンの公式	100,160
閾数	136	シンプソン法	160
間接アドレス指定	34	シンプレックスシステム	75
間接演算子	136	信頼性	62
ガント図	69	垂直分散処理システム	75
感熱式プリンタ	48	水平分散処理システム	75
記憶管理	60	スキャナ	71
機械語	62	スケジューリング機能	60
基数	2	スタック	37
基底アドレス指定	35	スタックポインタ	37
基底アドレスレジスタ	33	スタブ	63
基本計画	68	ストライピング	47
機密性	62	スパイウェア	74
キャッシュメモリ	41	スパイナルモデル	68
キャラクタ同期方式	53	スプリング機能	60
クイックソート法	152	スリーステートバッファ	31
クライアントサーバシステム	72	スループット	60
クラス	64	正規化	9
クラッカー	81	制御機能	106
クロックパルス	37	制御装置	33
継承	64	制御プログラム	60
ゲートウェイ	74	生体認証	74
けた落ち	7	セキュリティポリシー	81
結合テスト	69	セグメント	62
言語処理プログラム	60	セッション層	72
言語プロセッサ	62	絶対アドレス	34
原始プログラム	62	折衷テスト	69
公開鍵暗号方式	81	全加算回路	18
高水準言語	62	漸化式	89
構造化定理	69	先行制御方式	37
構造体	138	選択ソート（直接選択法）	82
公表権	80	セントロニクス	49
コーディング	69	全二重通信	52
五大機能	33	素因数分解	98
固定小数点	7	相対アドレス	34
コメント	177	相対アドレス指定	35
コンパイラ	62	挿入ソート（基本挿入法）	84
コンピュータウィルス	74	ソースプログラム	62
		ソート	82

即値アドレス指定	35	テスト	68	ハイパーテキスト	72
ソフトウェアパッケージ	66	手続き型言語	62	パイプライン方式	37
た行		デマルチプレクサ	29	パケット	54,73
ターミネータ	60	デュアルシステム	75	パケット交換網	54
ターンアラウンドタイム	60	デュプレックスシステム	75	パケットライティング	47
台形近似	101,159	電子掲示板	67	パスワード	74
ダイナミック方式	61	電子メール	72	バックアップ	81
タイムシェアリングシステム	75	転置式	81	ハッシング	63
タイムスライス方式	61	動的再配置	62	バッチ処理	75
対話型処理	75	トークンリング	74	バッファ	32
対話形式	48	度数分布	88	バブルソート(隣接交換法)	83
ダウンロード	72	特許権	80	パラレル伝送方式	52
タスク管理	60	ドットインパクトプリンタ	48	パリティ	10
タスクスケジューラ	60	ドット演算子	139	半加算回路	17
タッチスクリーン	48	トップダウンアプローチ	69	半二重通信	52
タブレット	48	ドメイン名	74	番兵	86
単純選択法	137	ドモルガン	13	汎用レジスタ	106
単体テスト	69	ドライバ	63	ピアツーピアシステム	72
単方向通信	52	トラック	44	ヒープソート法	154
逐次制御方式	37	トラックアットワنس	47	比較回路	18
逐次探索法	86,165	トラックボール	48	光ファイバ	57
チップセット	36	トランスポート層	72	ビッグバンテスト	69
チャタリング	32	トランステーラ	62	ビット	1,46
調歩同期方式	53	トレース	120	ビットスタッフィング	55
直接アドレス指定	34	ドロー系	78	ビット同期方式	52
直列伝送方式	52	な行		非手続き型言語	62
著作権	63	内部設計	68	ひまわり	71
著作隣接権	80	並べ換え	82	秘密鍵暗号方式	81
通信管理	60	二分法	104,157	表計算	66
通信管理システム	65	入出力管理	60	標本化	79
ディジタイザ	48	入出力チャネル	49	ピンポン伝送	59
ディジット	2	ニュースリーダ	72	ファームウェア	42
低水準言語	62	ニュートン法	103,158	ファイアウォール	81
ディスクアットワنس	47	入力機能	106	ファイル処理	141
ディスパッチャ	61	ネチケット	80	ファイルポインタ	141
データ管理	60	ネットニュース	72	プール代数	13
データ信号速度	52	ネットワーク層	72	フェイルセーフ	76
データ転送速度	52	ネットワークモデル	67	フェイルソフト	76
データベース	63	ノイマン型コンピュータ	33	フェッチサイクル	33
データベース管理システム	65	は行		フォールトレランス	76
データリンク層	72	バーコード	71	フォント	67
テクノ依存症	80	バーコードリーダ	48	復号化	81
テクノストレス	80	バージョンアップ	65	複製権	80
デコーダ	29	バイオメトリックス	81	符号化	79

プッシュダウンスタック	43	ボトムアップアプローチ	69	ラウンドロビン方式	61	
物理層	72	ホワイトボックステスト	69	ラストイン・ファーストアウト	43	
物理フォーマット	47	ま行			24	
浮動小数点	7	マウス	48	ラベル	119,177	
ブラウザ	72	マクロ	63	ランド	46	
フラグ同期方式	53	マクローリング数展開	99,156	リアルタイム処理	75	
フラグメンテーション	47	マスクROM	41	リーダ	60	
フラグレジスタ	33,106	魔方陣	97,162	リスト構造	164	
ブラックボックステスト	69	マルウェア	63	リピータ	74	
フラッシュメモリ	41	マルチタスク	61	リモートバッチ処理	75	
フリーソフトウェア	65	マルチプレクサ	29	量子化	79	
ブリッジ	74	マルチプロセッサシステム	75	リレーションナルデータベース	67	
フリップフロップ	23	マルティメディア	78	リンク	64	
プレゼンテーション	78	丸め誤差	7	ルータ	74	
プレゼンテーション層	72	ミドルウェア	65	レーザプリンタ	48	
プログラミング	68	ミラーリング	47	ローカルバッチ処理	75	
プログラム言語	62	無限小数	4	ロード	37	
プログラム設計	68	命令アドレスレジスタ	33	ロールイン／ロールアウト	62	
プログラム内蔵方式	33	命令実行段階	33	論理演算	11	
プロジェクト	48,79	命令デコーダ	33	論理代数	13	
ブロック間隔	45	命令取り出し段階	33	論理フォーマット	47	
ブロック同期方式	52	命令レジスタ	33	わ行		
ブロック分け	127	メーラー	72	ワークシート	67	
プロッタ	48	メソッド	64	ワードプロセッサ	66	
プロトコル	53,72	メッセージ	63,64	ワクチン	74	
プロタイプ宣言	136	メモリインタリープ	43	割込み処理	35	
プロタイプモデル	68	メモリセル	22			
文献検索	67	メンバ	138			
分散処理システム	75	目的プログラム	62			
分周器	24	文字の符号表	176			
平均故障間隔	75	モジュール	63			
平均修理時間	75	モデム	53			
並列伝送方式	52	モンテカルロ法	95,161			
ペイント系	78	や行				
ベーシック手順	53	有機EL	48			
ヘッダ情報	54	ユークリッド互除法	94			
ヘッド	44	有効アドレス	34			
変調速度	52	ユーティリティプログラム	60			
ポインタ	136	ら行				
ポインティングデバイス	48	ライタ	60			
ポート	36	ライトスルー	43			
保守容易性	61	ライトバック	43			
補数	5	ライトペン	48			
保全性	62	ラインプリンタ	48			

情報技術検定試験実施要項

- 1 主 催 社団法人 全国工業高等学校長協会
- 2 後 援 文部科学省
- 3 目 的 基礎的情報技術に関する知識と技能が習得されているかを検定する
- 4 実 施 月 6月、1月
- 5 会 場 受検希望校
- 6 受 檢 資 格 高等学校在校生・卒業者および受検校が受検を認めた者
- 7 検 定 方 法 筆記試験
- 8 検 定 料 3級 500円（税込み），2級 500円（税込み），1級 700円（税込み）
- 9 受 檢 手 続 所定の願書に記入の上検定料を添えて受検校の担当者に提出し、受検票を受け取る
- 10 合 格 基 準 3級・2級は70点以上、1級はI・IIともに70点以上
合格基準に達した場合は合格証書を授与する
- 11 特 別 表 彰 1級の合計点が190点以上の場合は特別表彰を行う
- 12 検 定 対 象 3級：情報処理技術入門者向け
2級：情報処理技術中級者向け
1級：「基本情報技術者試験」の受験者向け
- 13 各級検定試験内容・時間は「検定試験の内容」のとおり。
*詳しくは主催者にお問い合わせください。

検定試験の内容

3級検定の内容	1 コンピュータと社会 2 数の表現と論理 3 コンピュータの構成と利用 4 アルゴリズム 5 プログラム作成能力 (BASICまたはC言語のいずれかを選択)	試験時間 50分
	1 数の表現と演算 2 論理回路 3 ハードウェアの知識 4 ソフトウェアの基礎 5 コンピュータの利用 6 アルゴリズム 7 プログラム作成能力 (BASICまたはC言語のいずれかを選択)	
	I ハードウェアの基礎知識 1 数の表現と処理 2 コンピュータの基本回路 3 コンピュータの基本構成と各部の働き 4 通信 5 ソフトウェアの基礎 6 その他情報関連知識 (情報技術と社会、ネットワークなど)	
	II プログラミングの基礎知識 7 アルゴリズム 8 プログラムの作成能力 (C言語またはアセンブラー言語「CASL II」のいずれか選択)	

平成22年度 前期

文部科学省 後援

第44回 情報技術検定試験問題

1級 [I]

試験時間 50分

注意事項

1. 「始め」の合図があるまで、試験問題を開かないこと。
2. 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
3. 「始め」の合図があったら、試験問題を開き、試験をはじめること。
4. 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
5. 試験終了後、試験問題および解答用紙を提出すること。

社団法人 全国工業高等学校長協会

科	学年・組	受検番号	氏名

1 次の各間に答えなさい。

問 1 A, Bがそれぞれ $(0001\ 1011)_2$, $(0001\ 0000)_2$ であるとき, 次の計算を行いなさい。

$$(1) A \times B \div (32)_{10} = (\boxed{①})_2$$

$$(2) A \div B \div (16)_{10} = (\boxed{②})_{16}$$

問 2 次の空欄に最も適当な数字を入れなさい。

$125 - 50 = 55$ が成立するとき, $\boxed{③}$ 進法で計算している。

問 3 次の空欄に最も適当な数字を入れなさい。

10進数で表された 1 000 000 000 を, 2進数で表すと最低 $\boxed{④}$ ビットで表すことができる。

ただし, $\log_x Y = \frac{\log_2 Y}{\log_2 X}$ である。また, $\log_{10} 2 = 0.301$ とする。

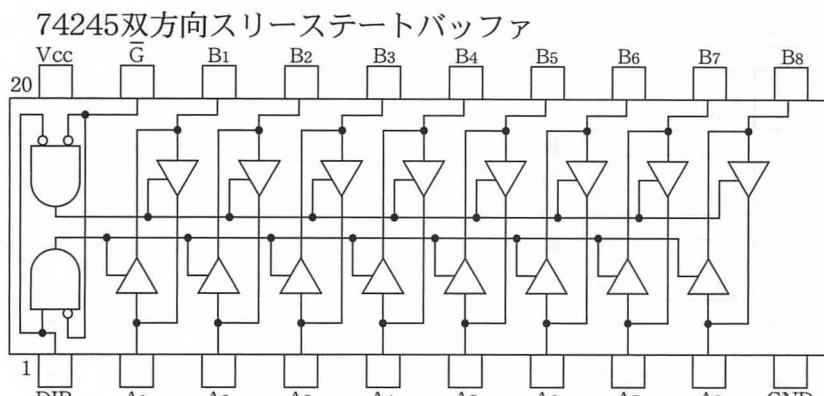
問 4 次の空欄に最も適当な数字を入れなさい。

16ビットで整数を表現する場合において, 負の数を 2 の補数で表すとき, 表現できる正の数の最大値は 32767 で, 負の数の最小値は $\boxed{⑤}$ である。

2 次の各間に答えなさい。

問1 図のTTL ICの真理値表を完成しなさい。ただし、①～⑤には、0または1のいずれかの論理が入るものとする。

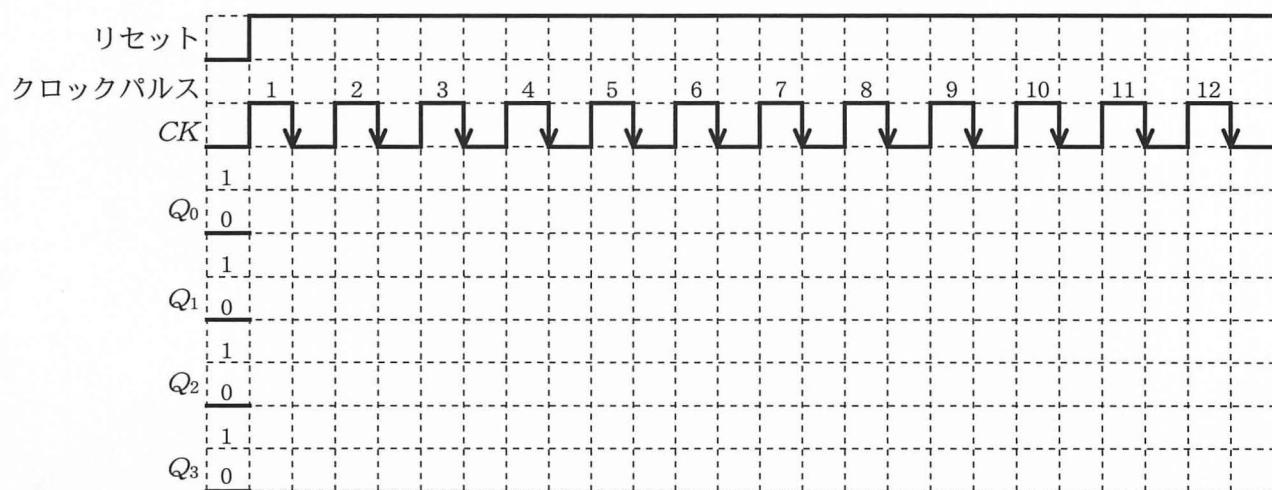
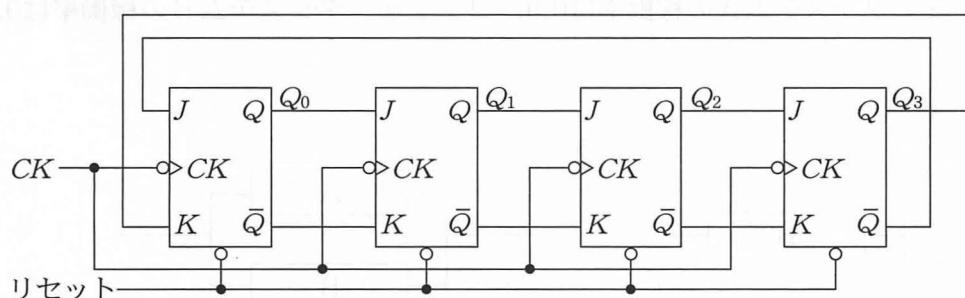
ただし、スリーステートバッファ回路は、出力が、ゲート入力により三つの状態をとる。ゲートが1のとき入力(1または0)がそのまま出力され、ゲートが0のとき入力に関係なく出力はハイインピーダンス(出力線が接続されていないのと同じ)になる。



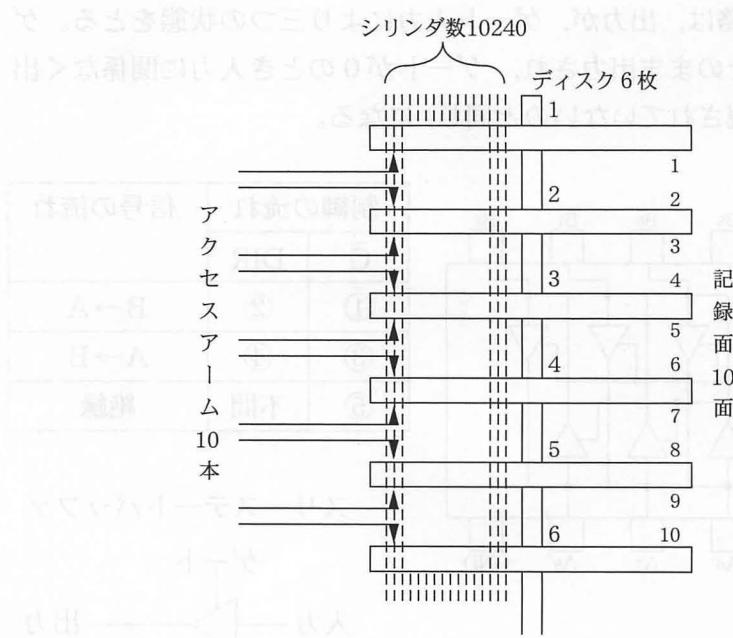
制御の流れ		信号の流れ
\bar{G}	DIR	
①	②	$B \rightarrow A$
③	④	$A \rightarrow B$
⑤	不問	絶縁

スリーステートバッファ
ゲート
入力 → 出力

問2 下図のJK-FFを4個接続したジョンソンカウンタと呼ばれる回路において、リセット信号で図のように $Q_0 \sim Q_3$ が0になった後のタイムチャートを描きなさい。



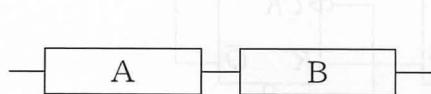
問3 図に示す磁気ディスク記憶装置において、シリング数が10240、1シリング当りのトラック数10、1トラック当りの記憶容量を512KBとすると、この記憶容量は何GBとなるか。
 ただし、1GB=1024MB、1MB=1024KBとする。



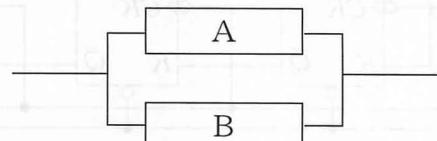
問4 次の図のようにコンピュータシステムを接続したとき、それぞれのシステム全体の稼働率を求めなさい。

ただし、コンピュータシステムAの稼働率は0.9、コンピュータシステムBの稼働率は0.8とする。

①



②



3 次の各間に答えなさい。

問1 次の説明に最も適切な語を解答群から選び、記号で答えなさい。

コンピュータの処理性能は、コンピュータの用途や使い方によって異なってくるため単純に比較することはできないので、比較の基準を定めることが大切である。

コンピュータは、①と呼ばれるパルス信号に同期して処理を行うため、このパルス信号の周波数が高い方が一般的に処理速度は速くなる。マイクロプロセッサが、ある1個の命令を実行するするために必要な①数を②と呼び、複数の命令の実行時間を計算する際などに利用される。

1秒間に実行できる命令数や演算数を処理性能の目安とする場合もある。1秒間に実行できる命令数を100万単位で表したものと③、1秒間に実行できる浮動小数点演算の回数を④という。

また、⑤と呼ばれる高速の記憶装置を効果的に使用することにより処理性能を向上させることもできる。

解答群

- ア. MIPS イ. FLOPS ウ. クロック エ. キャッシュメモリ オ. CPI

問2 次の文中の□に入るものを解答群から選び、記号で答えなさい。

(1) 一般に、記憶装置への情報の書き込み、読み取りには、ある程度の①を要する。記憶装置の同一場所に連続して書き込み（または読み取り）を行う場合、ある一つの書き込み（または読み取り）が始まってから次の書き込み（または読み取り）が始まるまでに要する時間を②と呼ぶ。また、制御装置が情報を要求した瞬間から、情報の受渡しが完了する瞬間までに要する時間を③という。

(2) 一般にMOS形メモリでは、蓄積された電荷が漏れて数ミリ秒後に記憶内容が失われる。そのため、この形のメモリでは記憶内容が失われないうちに再生する必要がある。それを④といい、そのようなメモリ素子を⑤形という。

解答群

- ア. サイクルタイム イ. 時間 ウ. リフレッシュ
エ. ダイナミック オ. アクセスタイム

4 次の各間に答えなさい。

問1 次の文章に当てはまる最も適当な語句を解答群から選び、記号で答えなさい。

マルチメディアでは、情報をデジタル化し、さらにデジタル化した情報に圧縮という操作を行ってデータ量を減らすことにより、伝達や保存が容易になる。このとき、圧縮されたデータを元の状態に戻すことを①といふ。①により、完全に元の情報に戻せる圧縮を②圧縮といい、完全に元には戻らず、情報が若干失われる圧縮を③圧縮といふ。静止画像の場合、②圧縮された画像データの例としては拡張子が④の画像形式が、③圧縮の例としては、拡張子が⑤の画像形式がある。

解答群

- | | | | | | |
|---------|--------|--------|---------|---------|--------|
| ア. MPEG | イ. MP3 | ウ. WAV | エ. JPEG | オ. TIFF | カ. CPU |
| キ. 伸張 | ク. 暗号化 | ケ. 量子化 | コ. 非可逆 | サ. 可逆 | |

問2 知的財産権などに関する次の記述中の①～⑤に入れるべき適当なものを解答群の中から選び、記号で答えなさい。

知的財産権には、大きく分けて、新しい発明の保護を目的とした①や、工業デザインを保護する②などの産業財産権と、思想・感情の創意的な表現を保護する目的の著作権の二つがある。著作権には、頒布権や③など多くの権利がある。また、著作物の実演を行う実演者の権利や放送事業者の権利は④で保護されている。

著作者の⑤や氏名表示権などは、人格権の一種であって財産権ではないが、著作者人格権として著作権などとともに扱われる。

解答群

- | | | | |
|--------|----------|----------|--------|
| ア. 肖像権 | イ. 著作隣接権 | ウ. 複製権 | エ. 公表権 |
| オ. 特許権 | カ. 意匠権 | キ. 実用新案権 | |

平成22年度 前期

文部科学省 後援

第44回 情報技術検定試験問題

1級 [II] C言語

試験時間 50分

注意事項

- 「始め」の合図があるまで、試験問題を開かないこと。
- 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
- 「始め」の合図があったら、試験問題を開き、最初に問題が①～⑤まであることおよび④と⑤がC言語の問題になっていることを確認した後に、試験を始める。
- 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
- 問題のアルゴリズムは、最適化されているものとする。また、問中のプログラムは、最も最適化されたアルゴリズムをもとに作成されているものとする。したがって、流れ図やプログラムにおいては、無駄な繰り返しや意味のない代入は行われていないものとする。
- 試験終了後、試験問題および解答用紙を提出すること。

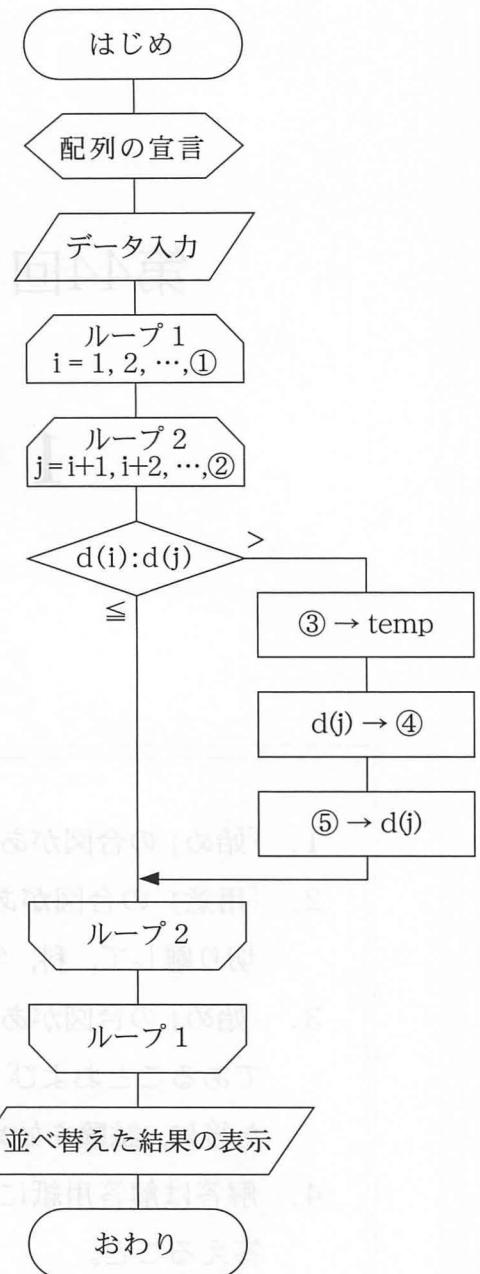
社団法人 全国工業高等学校長協会

科	学年・組	受検番号	氏名	
---	------	------	----	--

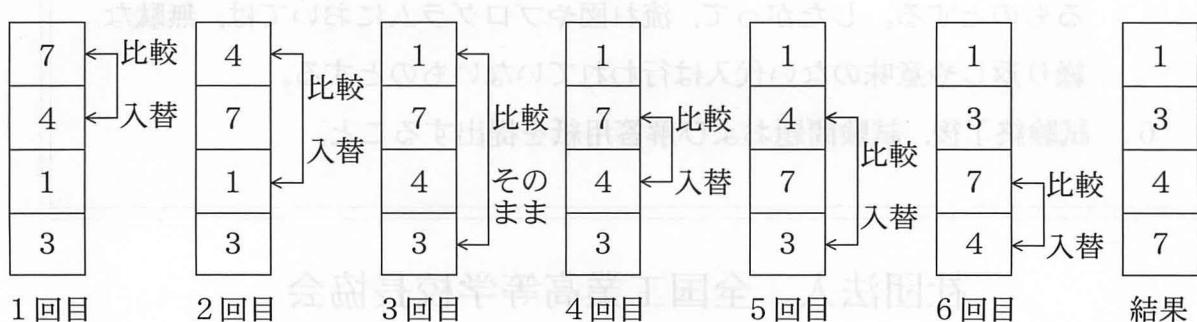
1 次の流れ図は、選択ソート（直接選択法）によって、n個のデータを昇順（小さいものから大きいものへの順）に並べ替えるアルゴリズムを示す。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

ただし、並べ替えの対象となるデータは配列 $d(1) \sim d(n)$ に入力されるものとする。また、データの入出力の繰り返し処理は、流れ図から省略してある。

選択ソートによる昇順の並べ替えの例を下図に示す。



選択ソートで昇順に並べ替える例



2 漸化式 $a_{k+2} = a_{k+1} + 2a_k$ ($k \geq 1$), $a_1 = 0$, $a_2 = 1$ で表される数列について、初項から第N項 ($N \geq 2$)までの和を求めるものである。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって完成しなさい。
ただし、ループの条件は終了条件である。

[例] Nが4の場合

$$a_1 = 0$$

$$a_2 = 1$$

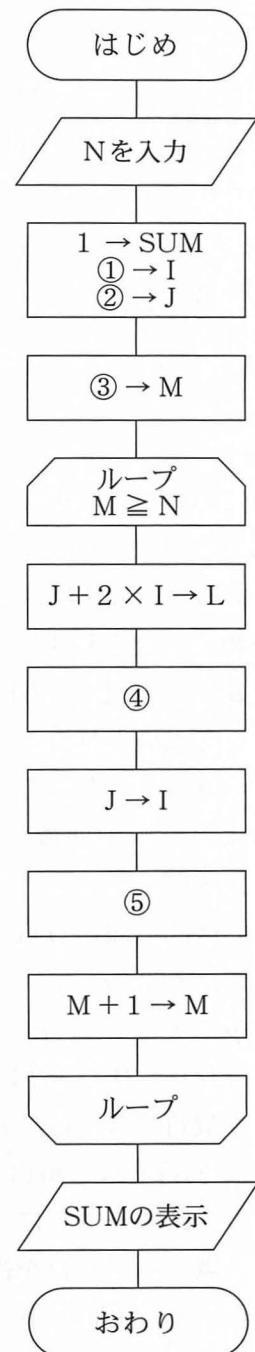
$$\begin{aligned} a_3 &= a_2 + 2a_1 \\ &= 1 + 2 \times 0 = 1 \end{aligned}$$

$$\begin{aligned} a_4 &= a_3 + 2a_2 \\ &= 1 + 2 \times 1 = 3 \end{aligned}$$

したがって、第4項までの総和は、

$$\begin{aligned} a_1 + a_2 + a_3 + a_4 \\ = 0 + 1 + 1 + 3 = 5 \end{aligned}$$

となる。



- 3 次の流れ図は、二分法によりある実数A ($A > 1$) の立方根を求めるものである。このとき、空欄①～⑤に入れるべきものを解答群から選び、その記号を答えなさい。ただし、ループ端の式は、繰り返しの終了条件である。

解答群

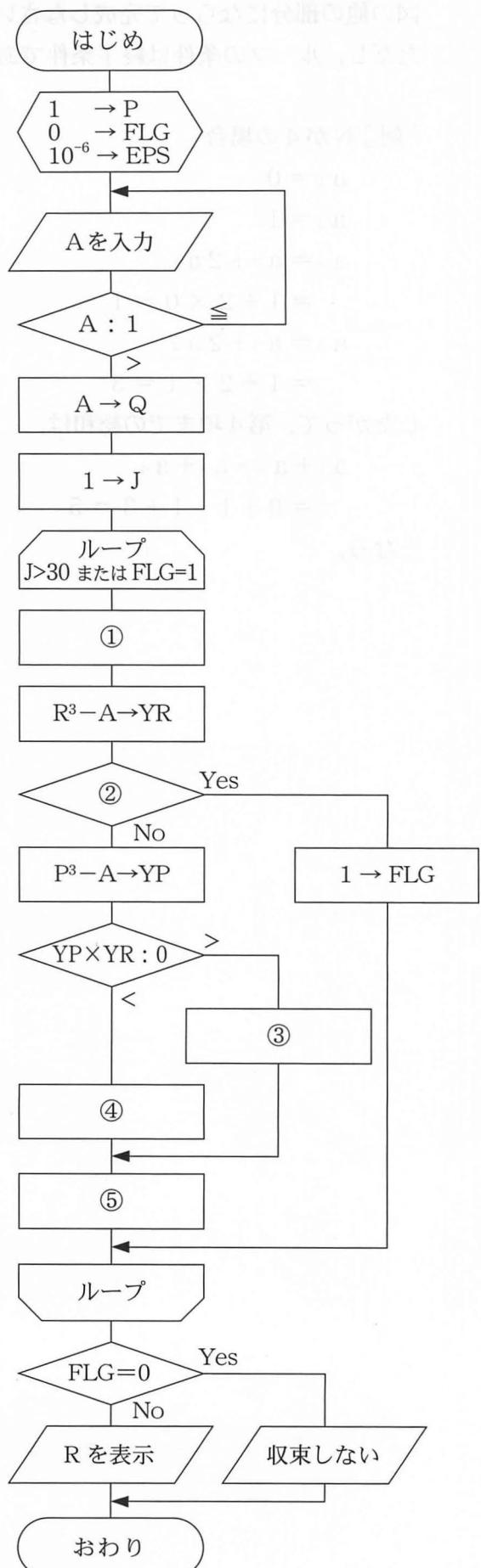
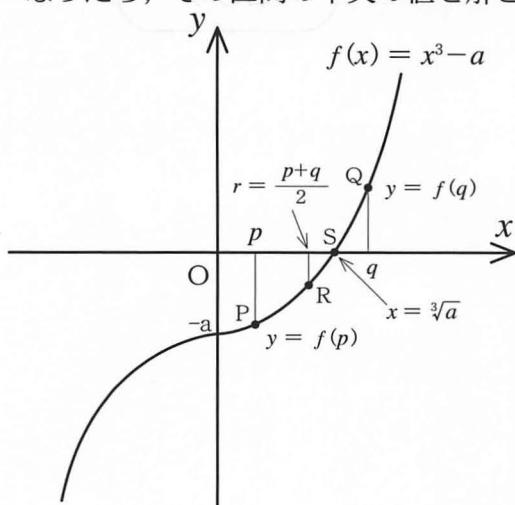
- | | |
|----------------------------------|------------------------------|
| ア. $Q - P < EPS$ または $YR = 0$ | |
| イ. $Q - P < EPS$ かつ $YR = 0$ | |
| ウ. $FLG = 1$ | エ. $FLG = 0$ |
| オ. $R \rightarrow Q$ | カ. $R \rightarrow P$ |
| キ. $\frac{P+Q}{2} \rightarrow R$ | ク. $P^3 - A \rightarrow R$ |
| ケ. $J + 1 \rightarrow J$ | コ. $FLG + 1 \rightarrow FLG$ |

考え方

下図において、曲線 $y = x^3 - a$ のグラフが、 x 軸の正の部分と交わる点 S の x 座標の値が求める立方根 $\sqrt[3]{a}$ である。二分法は、この点 S の左右で y の値の符号が変わることを利用する方法である。

実数 a の立方根は、方程式 $x^3 = a$ つまり $x^3 - a = 0$ の解として求められる。一般に方程式 $f(x) = 0$ の解は、 $y = f(x)$ のグラフと x 軸との交点であるので、下図のように解の前後で関数の値の符号は反転する。つまり、区間 $p < x < q$ において、 $f(p) \times f(q) < 0$ ならばこの区間の間に $f(x) = 0$ となる x が存在する。

そこで、区間 $[p, q]$ の中点 $r = (p + q)/2$ をとり、範囲を $p < x \leq r$ と $r \leq x < q$ に二分する。ここでもしも $f(r) = 0$ ならば r が解である。また、 $f(p)$ と $f(r)$ が異符号なら解は $p < x < r$ の区間にあり、 $f(r)$ と $f(q)$ が異符号なら解は $r < x < q$ の区間にある。解がある方の区間をさらに二分し、そのどちらかに解があるかを調べる。これを繰り返し、区間が十分小さくなったら、その区間の中央の値を解とする。



- 4 トランプのシャッフルと同様の操作をコンピュータで行いたい。52枚のトランプを1～52の数値に対応させる。a[1]～a[52]の配列に1～52の数値をランダムに入れることによりトランプを並べ替えたこととする。空欄①～④に入るべきものを記入し、プログラムを完成しなさい。ただし、プログラム中の「rand() % (k - 1) + 1」は、1以上k未満のランダムな整数を発生させることを表す。

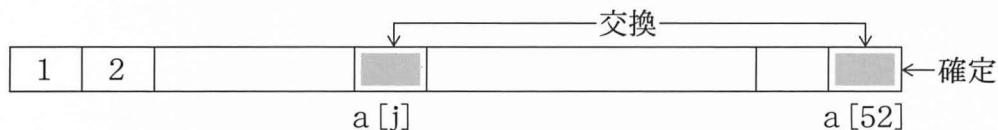
アルゴリズム

- (1) 1～52の値を配列a[1]～a[52]に順に格納する。

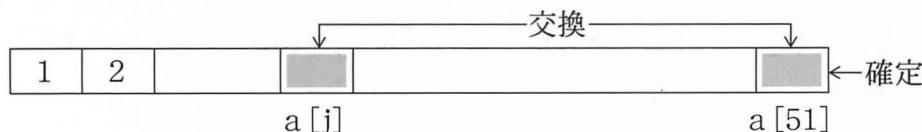
1	2	3			51	52
---	---	---	--	--	----	----

- (2) 1～51の範囲の乱数jを得て、これを添字とする配列a[j]とa[52]を交換する。

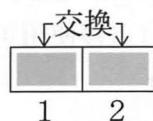
これにより、配列a[52]の値が決定するので、これを処理の対象外にする。



- (3) つぎに、1～50の範囲の乱数jを得て、これを添字とする配列a[j]とa[51]を交換する。これにより、配列a[51]の値が決定する。



- (4) 同様の処理を配列a[50]からa[2]まで繰り返す。



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    int a[53];
    int b, j, k;

    srand((unsigned)time( NULL ));

    for(k = 1; k <= 52; k++) {
        a[k] = [①];
    }

    for(k = 52; k >= [②]; k--) {
        [③] = rand() % (k - 1) + 1;
        b = a[k];
        a[k] = [④];
        a[j] = b;
    }

    printf("結果\n");
    for(k = 1; k <= 52; k++) {
        printf("%d ", a[k]);
    }
    printf("\n");

    return 0;
}
```

5 次のプログラムは、配列 data[] の文字列から、配列 pat[] で指定した文字列を逐次法により探索するプログラムである。プログラム中の①～⑤に入れるべき適切な語を解答群から選び、記号で答えなさい。

```
#include <stdio.h>

int main(void)
{
    char data[] = "C programming language developed for use with the Unix OS.";
    char pat[] = "Unix";
    char *dp, *pp;
    int i;

    dp = data;
    pp = [①];

    while (*dp != [②]) {

        for (i = 0; *(dp + i) == *(pp + i) [③] *(dp + i) != '\0'; i++) {
            ;
        }

        if (*(pp + i) [④] '\0') {
            break;
        }

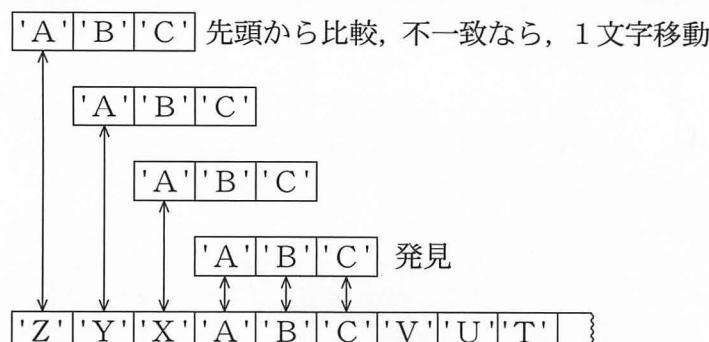
        dp++;
    }
    if (*(pp + i) [⑤] '\0') {
        printf("見つかりませんでした\n");
    }
    else {
        printf("見つかりました\n");
    }
    return 0;
}
```

解答群 (重複使用可)	
ア.	pat
イ.	data
ウ.	==
エ.	!=
オ.	'\0'
カ.	'\n'
キ.	
ク.	&&

ヒント

逐次探索法：

探査パターン文字列の先頭から末尾に向かって、順番に文字の比較を行い、不一致になると先に進む。



平成22年度 前期

文部科学省 後援

第44回 情報技術検定試験問題

1級 [II] CASL II

試験時間 50分

注意事項

- 「始め」の合図があるまで、試験問題を開かないこと。
- 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
- 「始め」の合図があったら、試験問題を開き、最初に問題が①～⑤まであることおよび④と⑤がCASL IIの問題になっていることを確認した後に、試験を始めること。
- 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
- 問題のアルゴリズムは、最適化されているものとする。また、問中のプログラムは、最も最適化されたアルゴリズムをもとに作成されているものとする。したがって、流れ図やプログラムにおいては、無駄な繰り返しや意味のない代入は行われていないものとする。
- 試験終了後、試験問題および解答用紙を提出すること。

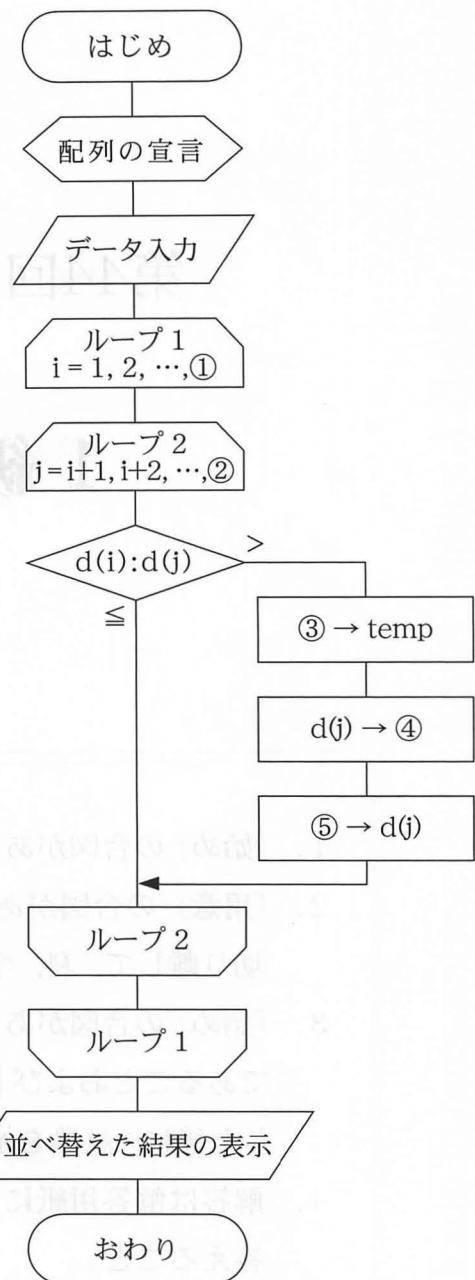
社団法人 全国工業高等学校長協会

科	学年・組	受検番号	氏名	
---	------	------	----	--

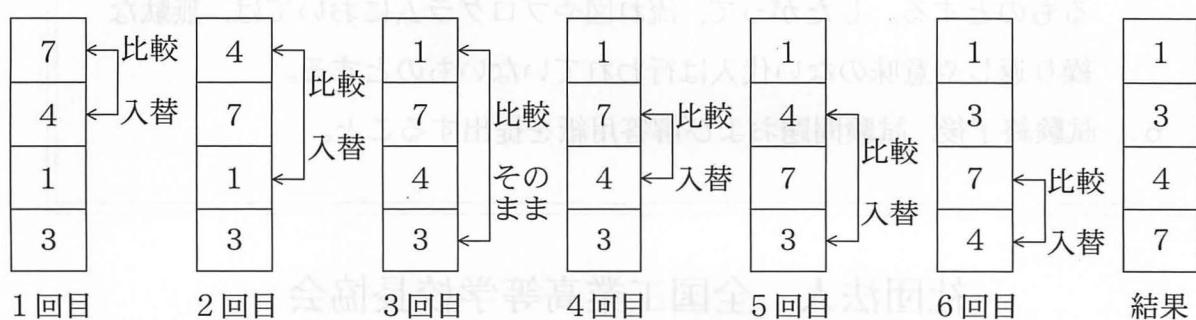
- 1 次の流れ図は、選択ソート（直接選択法）によって、n個のデータを昇順（小さいものから大きいものへの順）に並べ替えるアルゴリズムを示す。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

ただし、並べ替えの対象となるデータは配列 $d(1) \sim d(n)$ に入力されるものとする。また、データの入出力の繰り返し処理は、流れ図から省略してある。

選択ソートによる昇順の並べ替えの例を下図に示す。



選択ソートで昇順に並べ替える例



- 2 漸化式 $a_{k+2} = a_{k+1} + 2a_k$ ($k \geq 1$), $a_1 = 0$, $a_2 = 1$ で表される数列について、初項から第N項 ($N \geq 2$)までの和を求めるものである。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって完成しなさい。
- ただし、ループの条件は終了条件である。

[例] Nが4の場合

$$a_1 = 0$$

$$a_2 = 1$$

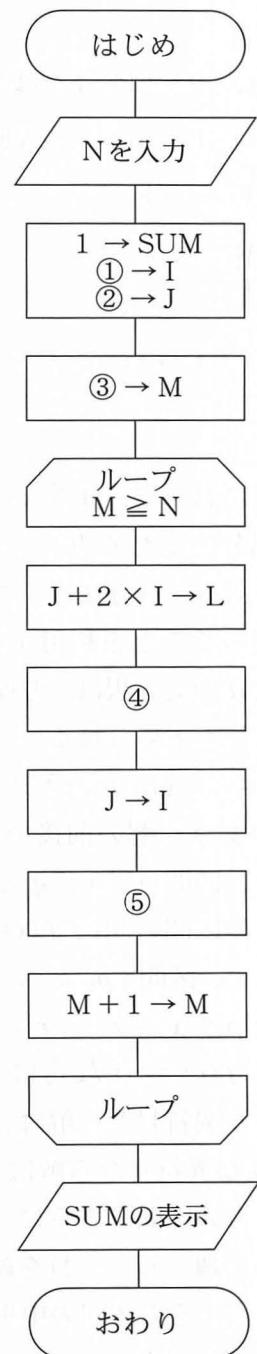
$$\begin{aligned} a_3 &= a_2 + 2a_1 \\ &= 1 + 2 \times 0 = 1 \end{aligned}$$

$$\begin{aligned} a_4 &= a_3 + 2a_2 \\ &= 1 + 2 \times 1 = 3 \end{aligned}$$

したがって、第4項までの総和は、

$$\begin{aligned} a_1 + a_2 + a_3 + a_4 \\ = 0 + 1 + 1 + 3 = 5 \end{aligned}$$

となる。



- 3 次の流れ図は、二分法によりある実数A ($A > 1$) の立方根を求めるものである。このとき、空欄①～⑤に入れるべきものを解答群から選び、その記号を答えなさい。ただし、ループ端の式は、繰り返しの終了条件である。

解答群

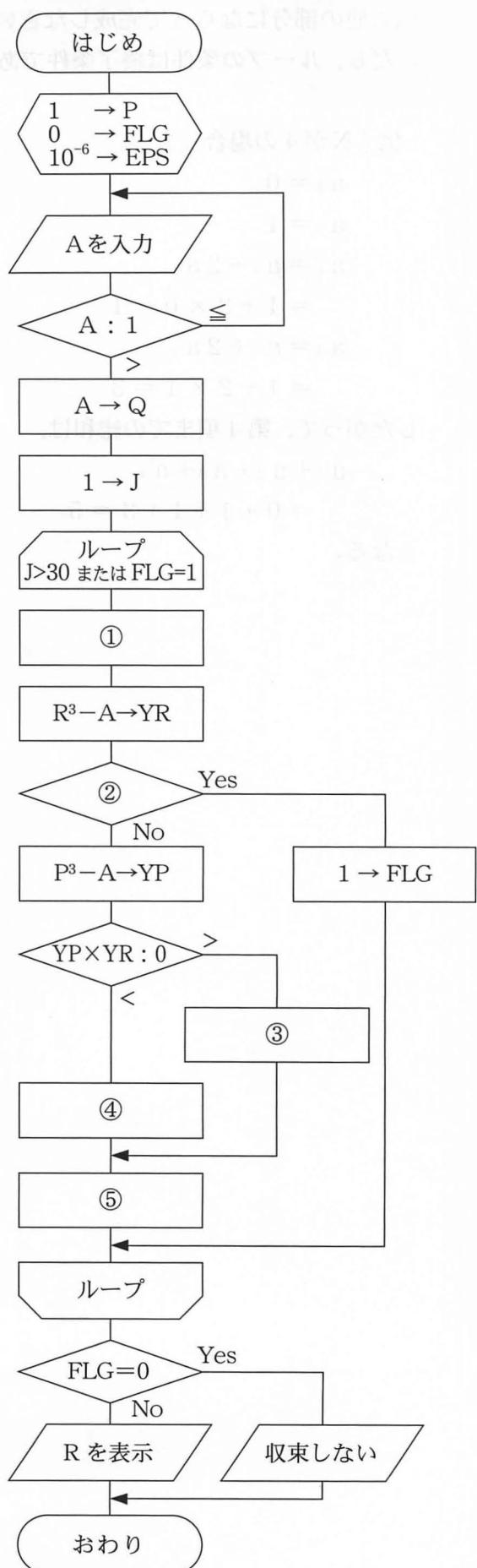
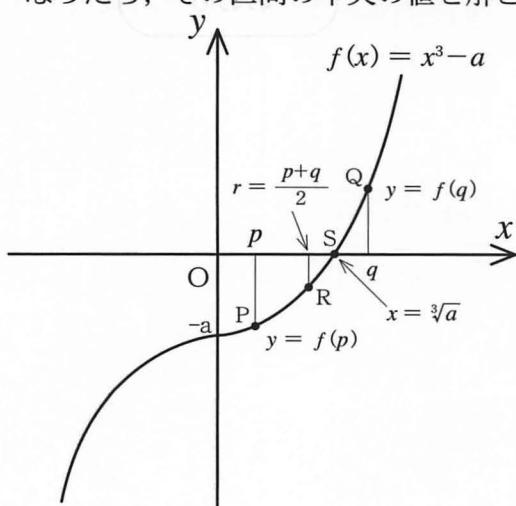
- | | |
|----------------------------------|------------------------------|
| ア. $Q - P < EPS$ または $YR = 0$ | |
| イ. $Q - P < EPS$ かつ $YR = 0$ | |
| ウ. $FLG = 1$ | エ. $FLG = 0$ |
| オ. $R \rightarrow Q$ | カ. $R \rightarrow P$ |
| キ. $\frac{P+Q}{2} \rightarrow R$ | ク. $P^3 - A \rightarrow R$ |
| ケ. $J + 1 \rightarrow J$ | コ. $FLG + 1 \rightarrow FLG$ |

考え方

下図において、曲線 $y = x^3 - a$ のグラフが、 x 軸の正の部分と交わる点 S の x 座標の値が求める立方根 $\sqrt[3]{a}$ である。二分法は、この点 S の左右で y の値の符号が変わることを利用する方法である。

実数 a の立方根は、方程式 $x^3 = a$ つまり $x^3 - a = 0$ の解として求められる。一般に方程式 $f(x) = 0$ の解は、 $y = f(x)$ のグラフと x 軸との交点である。下図のように解の前後で関数の値の符号は反転する。つまり、区間 $p < x < q$ において、 $f(p) \times f(q) < 0$ ならばこの区間の間に $f(x) = 0$ となる x が存在する。

そこで、区間 $[p, q]$ の中点 $r = (p + q)/2$ をとり、範囲を $p < x \leq r$ と $r \leq x < q$ に二分する。ここでもしも $f(r) = 0$ ならば r が解である。また、 $f(p)$ と $f(r)$ が異符号なら解は $p < x < r$ の区間にあり、 $f(r)$ と $f(q)$ が異符号なら解は $r < x < q$ の区間にある。解がある方の区間をさらに二分し、そのどちらかに解があるかを調べる。これを繰り返し、区間が十分小さくなったら、その区間の中央の値を解とする。



- 4 アセンブラー言語CASL IIが使用できる計算機COMET IIがある。次のプログラムの説明及びプログラムを読んで、プログラム中の①～④に入れるべき適切な語を補い、プログラムを完成しなさい。

[プログラムの説明]

このプログラムは、主プログラムから渡される二つの数値に、次の通りに対応する演算を施し、結果をGR 0に格納して主プログラムに返す副プログラムである。

- (1) 主プログラムから渡す二つの値は、それぞれGR 1, GR 2に格納してある。
- (2) 二つの数値に施すべき演算の種類は、表のように0～4にコード化してGR 3に格納してある。

(GR 3)	(GR 1), (GR 2)に対する演算
0	(GR 1)+(GR 2)
1	(GR 1)-(GR 2)
2	(GR 2)-(GR 1)
3	(GR 1)と(GR 2)との大きい方
4	(GR 1)と(GR 2)との小さい方

- (3) この副プログラムから戻るとき、汎用レジスタGR 1～GR 3は、呼び出されたときの内容のままとする。

[プログラム]

```

01 SUBRTN START
02          PUSH    0, GR 3
03          LD      ① ; 二つの数値の演算準備
04          LD      GR 3, LTBL, GR 3 ; } 該当演算処理
05          JUMP   0, GR 3 ; } に分岐
06 LB 0    ADDA   GR 0, GR 2 ; 加算
07          JUMP   LB 6
08 LB 1    SUBA   GR 0, GR 2 ; 減算 1
09          JUMP   LB 6
10 LB 2    LD      ②
11          SUBA   GR 0, GR 1 ; 減算 2
12          JUMP   LB 6
13 LB 3    CPA    GR 0, GR 2 ; 大きい値
14          JPL    ③
15          JUMP   ④
16 LB 4    CPA    GR 0, GR 2 ; 小さい値
17          JMI    LB 6
18 LB 5    LD      GR 0, GR 2
19 LB 6    POP    GR 3
20          RET
21 LTBL   DC     LB 0
22          DC     LB 1
23          DC     LB 2
24          DC     LB 3
25          DC     LB 4
26          END

```

- 5 アセンブラー言語 CASL II が使用できる計算機 COMET II がある。次のプログラムの説明及びプログラムを読んで、プログラム中の①～⑤に入れるべき適切な語を解答群から選び、記号で答えなさい。

[プログラム説明]

- (1) このプログラムは、入力装置から 1 レコードの文字データを入力し、英大文字はそのままで、それ以外の文字は “?” に変換して表示する。
- (2) 例 入力データ # J Y 1 2 K 結果の表示 ? J Y ?? K
- (3) 空のレコードが入力されるか又は EOF が検出されたとき、このプログラムは終了する。

[プログラム]

```

01  MAIN      START
02  INPUT     IN      BUF, LEN ; レコードの入力
03          LD      GR1, LEN
04          SUBA   GR1, =1 ; 終了判定
05          ①      FIN
06          LAD    GR0, 63  ; '?' の文字コード
07          LAD    GR2, 65  ; 'A' の文字コード
08          LAD    GR3, 91  ; 'Z' の 1 つ後の文字コード
09  LOOP      CPA    GR2, BUF, GR1
10          ②      CHG
11          CPA    GR3, BUF, GR1
12          ③      NEXT
13  CHG       ST     GR0, BUF, GR1
14  NEXT      SUBA   GR1, =1
15          JPL    LOOP
16          ④      LOOP
17          OUT   BUF, LEN ; 結果の表示
18          ⑤      INPUT
19  FIN       RET
20  BUF       DS    256
21  LEN       DS    1
22  END

```

— 解答群 (重複使用可) —

ア. JPL

イ. JMI

ウ. JZE

エ. JUMP

平成22年度 後期

文部科学省 後援

第45回 情報技術検定試験問題

1級 [I]

試験時間 50分

注意事項

1. 「始め」の合図があるまで、試験問題を開かないこと。
2. 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
3. 「始め」の合図があったら、試験問題を開き、試験をはじめるここと。
4. 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
5. 試験終了後、試験問題及び解答用紙を提出すること。

社団法人 全国工業高等学校長協会

科		学年・組		受検番号		氏名	
---	--	------	--	------	--	----	--

1 次の各間に答えなさい。

問1 16進数の1 AD 2について次の設間に答えなさい。

①10進数に変換しなさい。

②0.25倍した値を16進数で表しなさい。

問2 次の空欄に最も適切な数字を入れなさい。

$123 - 24 = 55$ が成立するとき、③進法で計算している。

問3 次の空欄に最も適切な数字を入れなさい。

10進数の1～10000をランプの点灯状態で表すものとする。10進数の1を表すときには1番目のランプを点灯、2を表すときには2番目のランプを点灯、……、10000を表すときには、10000番目のランプを点灯させると10000個のランプが必要になる。

2進数の0をランプ消灯、1をランプ点灯として10進数の1～10000を2進数で表すとき必要なランプは④個となり、10進数より2進数の方がランプの個数が少なくてよい。

ただし、 $\log_x Y = \frac{\log_2 Y}{\log_2 X}$ である。また、 $\log_{10} 2 = 0.301$ とする。

問4 次の空欄に最も適切な数字を入れなさい。

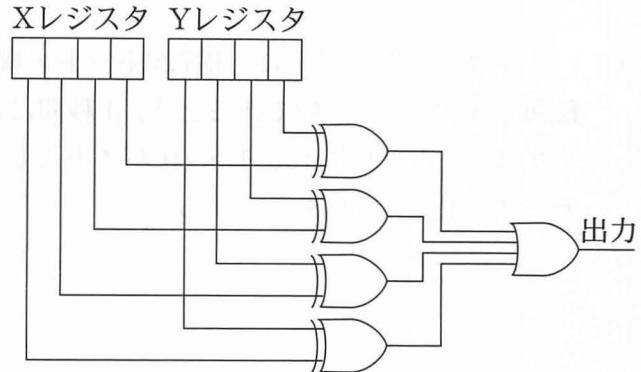
16ビットのデータを左に1ビットシフトすると、あふれが生じ、得られた値は16進数で ADB4 となった。元のデータを16進数で表すと⑤である。

2 次の各間に答えなさい。

問1 図の論理回路について、次の設間に答えなさい。

- (1) Xレジスタ、Yレジスタが表の状態のとき、出力端子の状態①、②はどうなるか。0または1で答えなさい。

Xレジスタ	Yレジスタ	出力端子
1011	0110	①
1001	1001	②



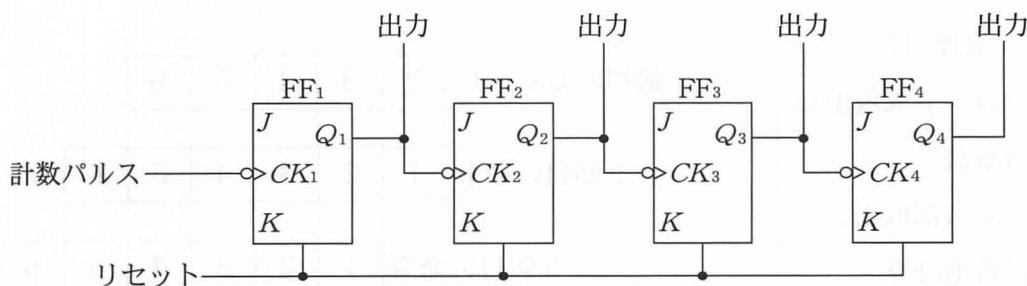
- (2) この論理回路の説明について、文中の空欄に入る最も適切な語を解答群から選び、記号で答えなさい。

Xレジスタ、Yレジスタのそれぞれの対応するビットが EX-OR 回路で接続されており、各ビットごとの不一致を検出する。更にこれらの各ビットごとの出力が OR 回路に入力されているため、OR回路のすべての入力(すべてのEX-OR回路の出力)が [③] のときのみ出力が [④] であるので、対応する全ビットが一致しているとき、[④] を出力する。すなわち、[⑤] 回路であることがわかる。

解答群 (重複利用可)

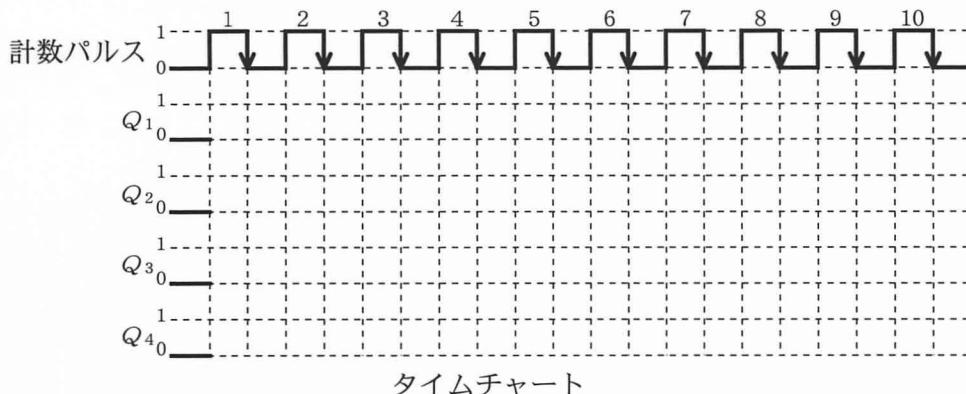
- ア. 0 イ. 1 ウ. 一致検出 エ. パリティーチェック オ. キャリー検出 ハ. 比較

問2 図のような4ビットの非同期式純2進カウンタのタイムチャートを完成しなさい。



(入力J, Kは両方とも1である)

(FFは入力パルスの立下りで動作し、カウンタは、最初リセット状態にあるものとする)



問3 次の文章を読んで、下の設問に答えなさい。

データ信号速度が100Mbpsの通信回路で画像データを伝送したい。ただし、bpsとは、1秒間に伝送可能なビット数である。また、 $1\text{ Mbps} = 1 \times 10^6\text{ bps}$ とする。

- (1) パソコンのモニターに、横720ドット×縦480ドットで256色の動画を表示させたい。この動画が30フレーム/秒であるとき、1秒間に表示される動画データの量は約何Mバイトか。ただし、1Mバイト = 1×10^6 バイトとし、小数第1位を四捨五入して求めなさい。また、データは圧縮しないものとする。



- (2) (1) の動画データを $1/5$ に圧縮して伝送するには何秒かかるか。ただし、回線利用率を40%とする。

各々のアドレスが複数の出力端子を持つ複数のデータバスによって接続され、各々のアドレスも複数の出力端子を持つ複数のデータバスによって接続される。各々のアドレスバスと各々のデータバスは、各々のアドレス端子と各々のデータ端子が一一対応する。各々のアドレス端子と各々のデータ端子は、各々のアドレスバスと各々のデータバスによって接続される。

問4 次のようなパイプライン処理で命令を実行するコンピュータで、10個の命令を実行する実行時間は何ナノ秒か。ただし、各ステップの実行時間は全て10ナノ秒とし、パイプライン処理の実行を乱す要因はないものとする。

ステップ	処理内容
1	命令コード部取出し
2	命令解読
3	アドレス部取出し
4	実行番地計算
5	データ取出し
6	演算実行



3 次の各間に答えなさい。

問1 次の文章の空欄に入る最も適切な語句を解答群から選び、記号で答えなさい。

パソコンなどで使用される記憶媒体には、情報を読み出すだけのものや、読み書きすることが可能なものがある。レーザ光を用いてプラスチック盤にプレスされた①を読みとるDVD-ROMは、情報を読み出すだけの記憶媒体である。レーザ光を用いて記録層を②に変化させて情報を記録する方式を用いるDVD-RWは、情報を書き換えることが可能である。

情報の書き込みはできるが書き換えのできないDVD-Rは、③をレーザー光で焼き付けて情報を記録している。また、④で情報を記憶するFDや、記録面にレーザ光を照射し⑤で磁化の方向を変えることでデータを記録するMOは、情報を書き換えることができる記憶媒体である。

解答群

- | | | |
|------------|---------|-------------------------|
| ア. 有機色素 | イ. 磁気と熱 | ウ. 結晶状態または非結晶（アモルファス）状態 |
| エ. 磁気 | オ. 光 | カ. 正電荷と負電荷 |
| キ. ピットとランド | | |

問2 次の説明に最も適切な語を解答群から選び、記号で答えなさい。

- ①カード表面に磁性体を持ち磁気記録する。
- ②コンピュータに直列伝送方式で周辺装置を接続するための規格。
- ③集積回路とアンテナをカード内部に内蔵し、非接触でデータの読み書きができる。
- ④二次元の模様でデータを記憶する。
- ⑤赤外線を用いてノートパソコンとプリンタなど近距離での通信が可能である。

解答群

- | | | | | | |
|---------|---------|------------|----------|----------|--------|
| ア. IrDA | イ. RFID | ウ. UVEPROM | エ. QRコード | オ. 磁気カード | カ. USB |
|---------|---------|------------|----------|----------|--------|

4 RASISに関する次の記述に当てはまる最も適切な語句を解答群から選び、記号で答えなさい。

- (1) R は、reliabilityの頭文字で、①と呼ばれる。これは、システムが安定して動作する目安で、指標として故障から故障までの間隔の平均である②が用いられる。
- (2) A は、availabilityの頭文字で、③と呼ばれる。これは、コンピューターが正常に動作している割合を示し、指標として④が用いられる。
- (3) 一つ目の S は、serviceabilityの頭文字で、⑤と呼ばれる。システムに障害が発生したときの修理のしやすさを示し、指標として修理に要する時間の平均である⑥が用いられる。
- (4) I は、integrityの頭文字で、⑦と呼ばれる。これは、ハードウェアの故障およびソフトウェアの異常や故意によるデータの破壊を防止できること。また、もしもデータが破壊されても容易に修復できること。
- (5) 最後の S は、securityの頭文字で、⑧と呼ばれる。これは、情報の漏洩や破壊などを防ぎ、データの保護や機密保持を行うこと。
- (6) システムに障害が発生したときに、正常な動作を保ち続ける能力をフォールトトレランスという。これには、システムの一部が故障しても、全体としての能力を下げても引き続き動作を可能にする⑨と、安全を重視し、故障による被害を最小限に抑えることを重視した動作を行う⑩がある。

解答群

- | | | | | |
|------------|------------|--------|---------|---------|
| ア. 保守容易性 | イ. 汎用性 | ウ. 機密性 | エ. 同時性 | オ. 信頼性 |
| カ. 保全性 | キ. 可用性 | ク. 稼働率 | ケ. MTTR | コ. MTBF |
| サ. フェイルセーフ | シ. フェイルソフト | | | |

平成22年度 後期

文部科学省 後援

第45回 情報技術検定試験問題

1級 [II] C言語

試験時間 50分

注意事項

- 「始め」の合図があるまで、試験問題を開かないこと。
- 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
- 「始め」の合図があったら、試験問題を開き、最初に問題が①～⑤まであること及び④と⑤がC言語の問題になっていることを確認した後に、試験を始めること。
- 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
- 問題のアルゴリズムは、最適化されているものとする。また、問中のプログラムは、最も最適化されたアルゴリズムをもとに作成されているものとする。したがって、流れ図やプログラムにおいては、無駄な繰り返しや意味のない代入は行われていないものとする。
- 試験終了後、試験問題及び解答用紙を提出すること。

社団法人 全国工業高等学校長協会

科	学年・組	受検番号	氏名	
---	------	------	----	--

- 1 2分探索法のアルゴリズムを示す次の流れ図の中の空欄①～④に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

ただし、検索の対象となるデータは、配列 $d(1) \sim d(n)$ に格納されており、昇順（小さいものから大きいものへの順）に並んでいるものとする。また、検索対象データは変数 key に入力されるものとする。

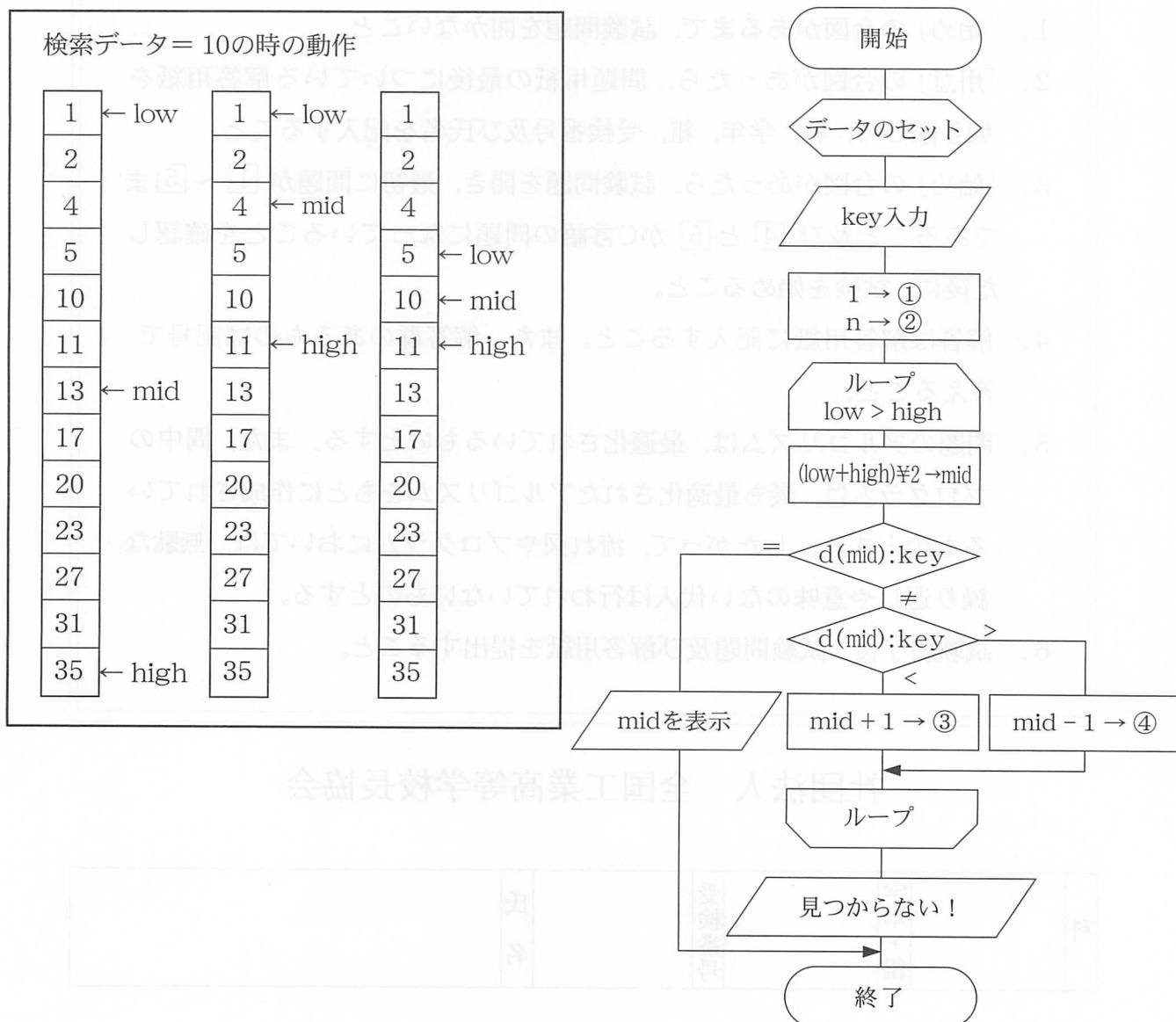
注意：

流れ図中の「 $A \% B$ 」は、 A を B で割った商と余りのうちの、商を求める演算を表す。

また、ループ開始端の式は、繰り返しの終了条件を表すものとする。

[参考]

2分探索法は、データがソートされて順番に並んでいるときに有効な探索法である。いま、データが昇順に並んでいるものとする。探索範囲の上限を $high$ 、下限を low とするとき、 $mid = (low + high)/2$ （小数点以下切捨て）の位置のデータとキーデータを比較する。キーデータの方が大きければ、キーデータは mid より上にあり、キーデータの方が小さければ、キーデータは mid より下にある。これを元に、新たな low と $high$ の値を設定して比較を繰り返す。下に、例を図示する。



- 2 次の流れ図は、入力された整数 N が連續した二つ以上の正の整数の和として表すことができるかどうか調べるものである。存在する場合はその連續する整数のパターンを全て出力し、無い場合は何も出力しない。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

[例] 整数21ならば、

$$\begin{aligned} 21 &= 1 + 2 + 3 + 4 + 5 + 6 \\ &= 6 + 7 + 8 \\ &= 10 + 11 \end{aligned}$$

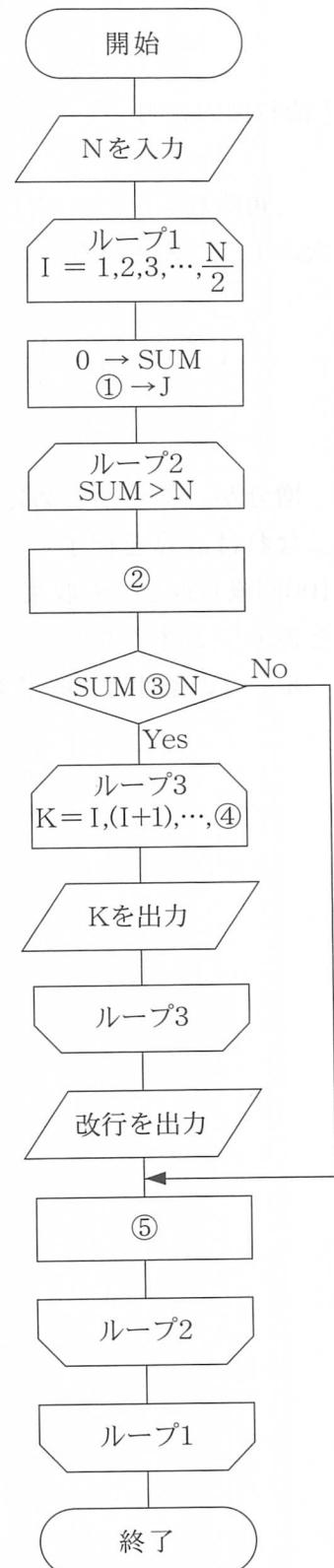
の3パターンである。

[例] 整数8ならば、一つもない。

注)

$\frac{N}{2}$ は小数点以下は切り捨てとする。

ループ2の条件は終了条件である。



- 3 次の流れ図は、マクローリン級数展開をもつて、三角関数 $\sin x$ を求めて表示するプログラムである。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

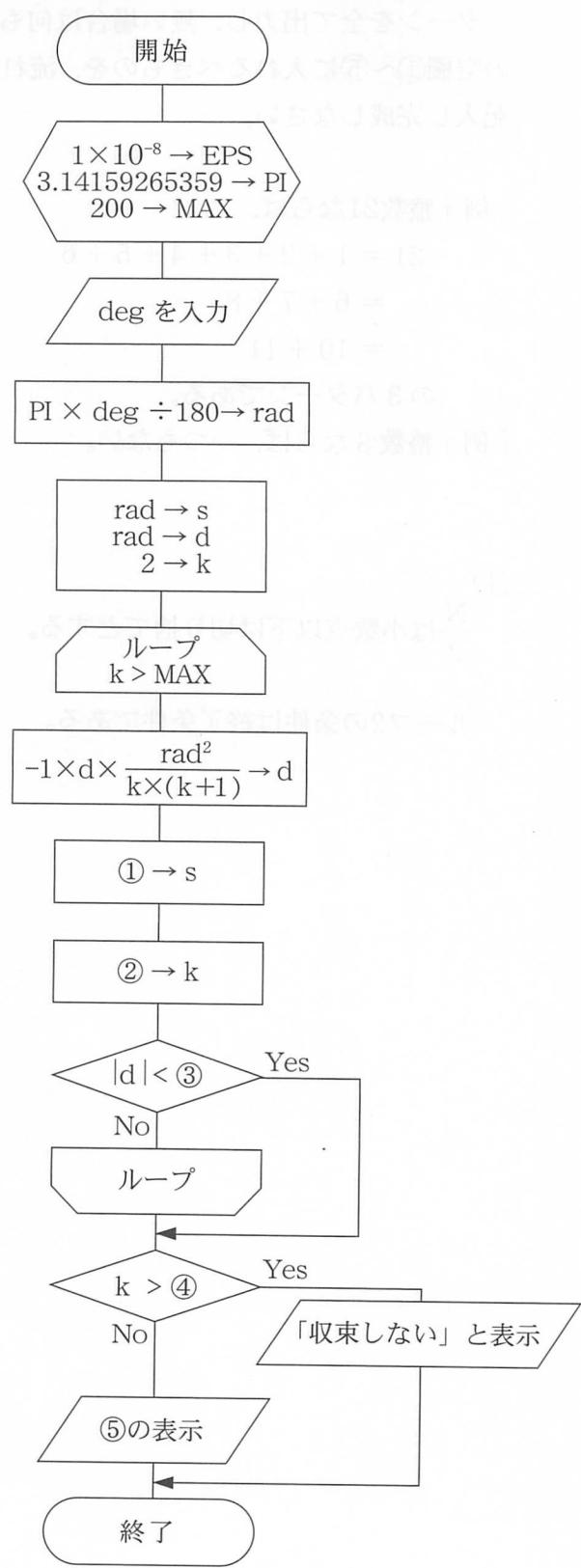
[流れ図の説明]

三角関数 $\sin x$ をマクローリン級数に展開すると、次のように表される。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

増分が、あらかじめ定められた値 ϵ より小さくなれば計算を終了して結果を表示する。また、100回繰り返しても収束しなければ、「収束しない」と表示して終了する。

ループの条件は、終了条件である。



- 4 次のプログラムは、金額を入力すると、必要な紙幣と硬貨の枚数を計算して表示するものである。空欄①～④に入れるべきものを記入し、プログラムを完成しなさい。ただし、紙幣は1万円、5千円、千円、硬貨は500円、100円、50円、10円、5円、1円とし、配列 $m[0] \sim m[8]$ に数値が格納されているものとする。また、必要な紙幣と硬貨の枚数は配列 $n[0] \sim n[8]$ に格納されるものとする。ただし、0円以下の金額を入力するとプログラムは終了する。

[ヒント]

例えば、入力された金額が147,326円とすると、まずこれを10,000で割り、商を求めるとき14で、余り7,326となり、1万円札が14枚必要であることがわかる。次に147,326円から140,000円を引いた余り7,326円について、こんどは5,000円以下の必要枚数を求めていけばよい。

```
#include <stdio.h>

int main(void)
{
    int i, m[9] = {10000, 5000, 1000, 500, 100, 50, 10, 5, 1}, n[9];
    long t;

    for (;;) {
        printf("金額を入れてください[0以下で終了]:");
        scanf("%ld", &t);
        if (t <= 0) break;

        for (i = 0; i <= 8; i++) {
            if (t >= [①]) {
                n[i] = [②];
                t = [③];
            }
            else {
                n[i] = [④];
            }
        }

        for (i = 0; i <= 8; i++) {
            printf("%5d円%5d枚¥n", m[i], n[i]);
        }
    }

    return 0;
}
```

実行結果

金額を入れてください[0以下で終了]:147326	
10,000円	14枚
5,000円	1枚
1,000円	2枚
500円	0枚
100円	3枚
50円	0枚
10円	2枚
5円	1枚
1円	1枚
金額を入れてください[0以下で終了]:	

配列と格納される数値

配列	数値	配列	格納されるデータ
$m[0]$	10,000	$n[0]$	1万円札の枚数
$m[1]$	5,000	$n[1]$	5千円札の枚数
$m[2]$	1,000	$n[2]$	千円札の枚数
$m[3]$	500	$n[3]$	500円硬貨の枚数
$m[4]$	100	$n[4]$	100円硬貨の枚数
$m[5]$	50	$n[5]$	50円硬貨の枚数
$m[6]$	10	$n[6]$	10円硬貨の枚数
$m[7]$	5	$n[7]$	5円硬貨の枚数
$m[8]$	1	$n[8]$	1円硬貨の枚数

C 言語選択用

[5] 次のプログラムは、与えられた複数の文字列を昇順に表示するものである。空欄①～⑤に当てはまる適切な語を解答群から選び、記号で答えなさい。

```
#include <stdio.h>
#include <string.h>
void disp(char *[]);
void sort(char *[]);
void swap(char *[], int, int);
int main(void)
{
    char *country[5] = {"Japan", "Uruguay", "Germany", "Netherlands", "Spain"};
}

```

```
    [①];
    disp(country);
    return 0;
}

void sort(char [②])
{
    int i, j;
```

```
    for (i = 0; i < 4; i++) {
        for (j = [③]; j < 5; j++) {
            if (strcmp(c[i], c[j]) > 0) {
                [④];
            }
        }
    }
}
```

```
void swap(char *c[], int i, int j)
{
```

```
    char *temp;

    temp = c[i];
    c[i] = c[j];
    c[j] = temp;
}
```

```
void disp(char *c[])
{
    int i;

    for (i = 0; i < 5; i++) {
        printf("%s\n", [⑤]);
    }
}
```

[ヒント] strcmp 関数について

書式: int strcmp(char *str1, char *str2);

strcmp 関数は文字列を比較する関数である。
第1引数と第2引数の文字列が等しい場合は値0を返す。

また、第1引数のほうが大きい場合は、正の値を返し、逆に、第2引数のほうが大きい場合は負の値を返す。この大小は辞書順（文字コード順）となる。

実行結果

```
Germany
Japan
Netherlands
Spain
Uruguay
```

解答群

- ア) swap(c, i, j)
- イ) swap(c[i], c[j])
- ウ) c[]
- エ) *c[]
- オ) c[i]
- カ) c[j]
- キ) sort(country)
- ク) sort(country[])
- ケ) i
- コ) i + 1

平成22年度 後期

文部科学省 後援

第45回 情報技術検定試験問題

1級 [II] CASL II

試験時間 50分

注意事項

1. 「始め」の合図があるまで、試験問題を開かないこと。
2. 「用意」の合図があったら、問題用紙の最後についている解答用紙を切り離して、科、学年、組、受検番号及び氏名を記入すること。
3. 「始め」の合図があったら、試験問題を開き、最初に問題が①～⑤まであること及び④と⑤がCASL IIの問題になっていることを確認した後に、試験を始めること。
4. 解答は解答用紙に記入すること。また、解答群のあるものは記号で答えること。
5. 問題のアルゴリズムは、最適化されているものとする。また、問中のプログラムは、最も最適化されたアルゴリズムをもとに作成されているものとする。したがって、流れ図やプログラムにおいては、無駄な繰り返しや意味のない代入は行われていないものとする。
6. 試験終了後、試験問題及び解答用紙を提出すること。

社団法人 全国工業高等学校長協会

科	学年・組	受検番号	氏名
---	------	------	----

- 1 2分探索法のアルゴリズムを示す次の流れ図の中の空欄①～④に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

ただし、検索の対象となるデータは、配列 $d(1) \sim d(n)$ に格納されており、昇順（小さいものから大きいものへの順）に並んでいるものとする。また、検索対象データは変数 key に入力されるものとする。

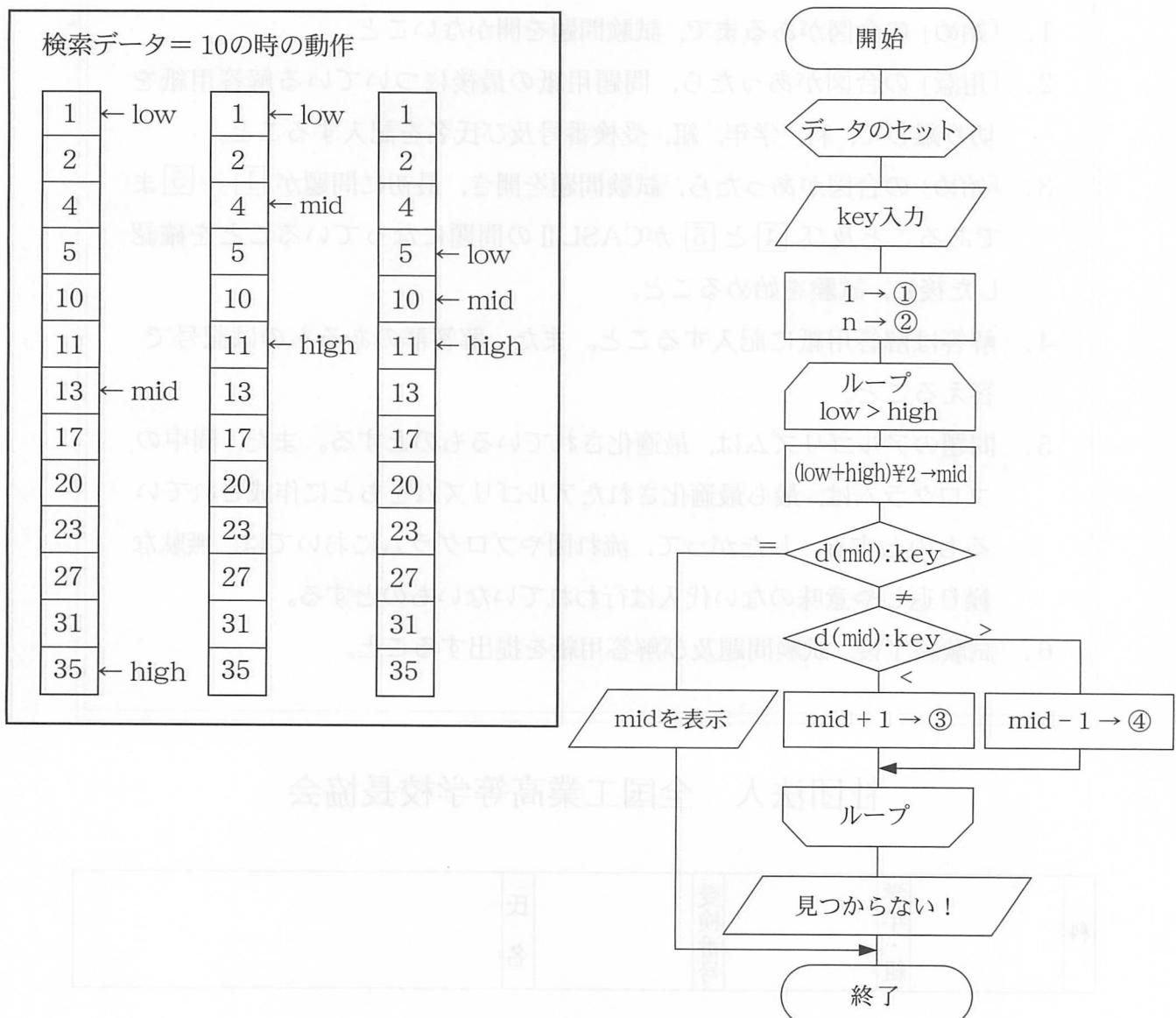
注意：

流れ図中の「 $A \div B$ 」は、 A を B で割った商と余りのうちの、商を求める演算を表す。

また、ループ開始端の式は、繰り返しの終了条件を表すものとする。

[参考]

2分探索法は、データがソートされて順番に並んでいるときに有効な探索法である。いま、データが昇順に並んでいるものとする。探索範囲の上限を $high$ 、下限を low とするとき、 $mid = (low + high)/2$ （小数点以下切捨て）の位置のデータとキーデータを比較する。キーデータの方が大きければ、キーデータは mid より上にあり、キーデータの方が小さければ、キーデータは mid より下にある。これを元に、新たな low と $high$ の値を設定して比較を繰り返す。下に、例を図示する。



2 次の流れ図は、入力された整数 N が連續した二つ以上の正の整数の和として表すことができるかどうか調べるものである。存在する場合はその連續する整数のパターンを全て出力し、無い場合は何も出力しない。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

[例] 整数21ならば、

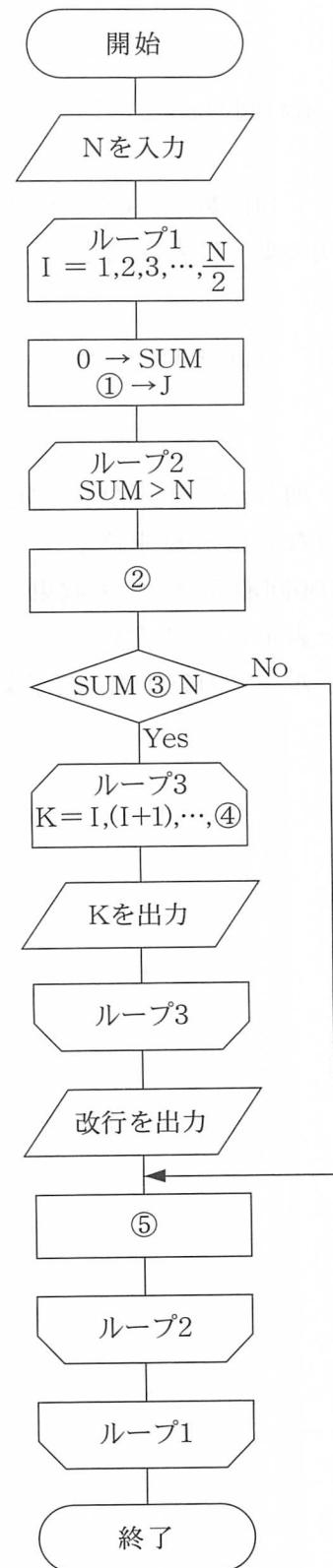
$$\begin{aligned} 21 &= 1 + 2 + 3 + 4 + 5 + 6 \\ &= 6 + 7 + 8 \\ &= 10 + 11 \end{aligned}$$

の3パターンである。

[例] 整数8ならば、一つもない。

注) $\frac{N}{2}$ は小数点以下は切り捨てとする。

ループ2の条件は終了条件である。



- 3 次の流れ図は、マクローリン級数展開をもつて、三角関数 $\sin x$ を求めて表示するプログラムである。流れ図の中の空欄①～⑤に入れるべきものを、流れ図の他の部分にならって記入し完成しなさい。

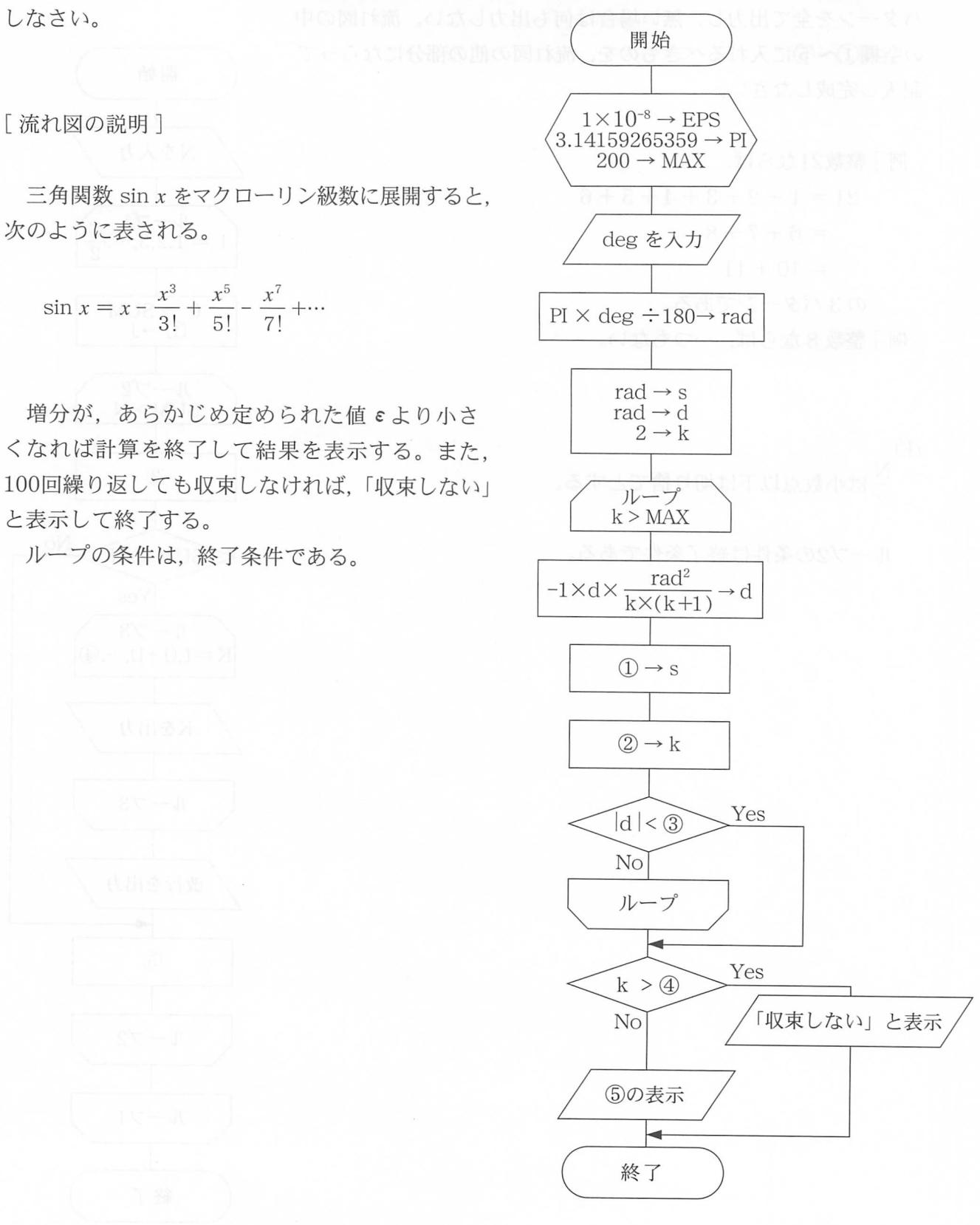
[流れ図の説明]

三角関数 $\sin x$ をマクローリン級数に展開すると、次のように表される。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

増分が、あらかじめ定められた値 ϵ より小さくなれば計算を終了して結果を表示する。また、100回繰り返しても収束しなければ、「収束しない」と表示して終了する。

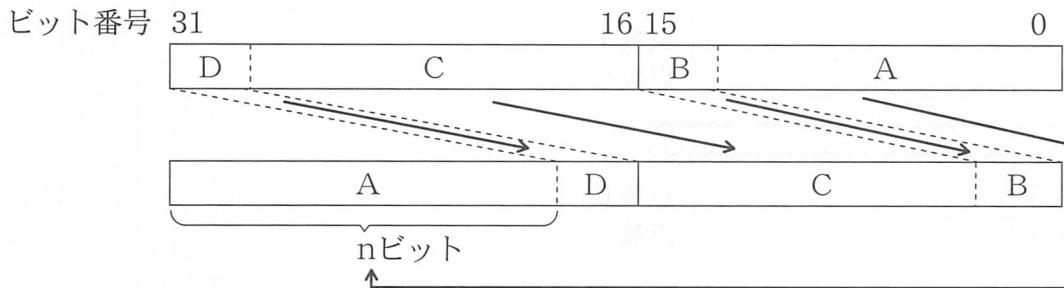
ループの条件は、終了条件である。



- 4 アセンブラー言語CASL IIが使用できる計算機COMET IIがある。次のプログラムの説明及びプログラムを読んで、プログラム中の①～④に入れるべき適切な語を補い、プログラムを完成させなさい。

[プログラムの説明]

連続する2語を32ビットのビット列と見なす。このビット列を指定されたビット数nだけ右に循環シフトする副プログラムSHIFTである。右に循環シフトするとは、図に示すように、シフトの結果、右にはみ出したビット列を左端に格納することを言う。なお図において、ビット0～15のビット列が第1語、16～31ビット列が第2語になる。



- (1) ビット列の先頭番地はGR 1に、シフトするビット数nはGR 2に格納され、主プログラムから渡される。
- (2) シフトした結果は、もとの領域に格納される。
- (3) $0 \leq n \leq 16$ とする。

[プログラム]

```

01 SHIFT      START
02           RPUSH
03           LD      GR 0, 0, GR 1
04           CALL    DIVIDE      ; 第1語を二つの部分に分ける
05           LD      GR 4, GR 0
06           LD      ①, GR 3
07           LD      GR 0, 1, GR 1
08           CALL    DIVIDE      ; 第2語を二つの部分に分ける
09           OR      GR 3, ②
10           ST      GR 3, 0, GR 1
11           OR      GR 0, GR 5
12           ST      GR 0, 1, GR 1
13           RPOP
14           RET
15           END
16 ;
17 DIVIDE     START
18           ST      GR 2, SAVE
19           LD      GR 3, ③
20           SRL    GR 0, 0, GR 2
21           LAD    GR 2, 16
22           SUBA   GR 2, ④
23           SLL    GR 3, 0, GR 2
24           LD      GR 2, SAVE
25           RET
26 SAVE       DS      1
27           END

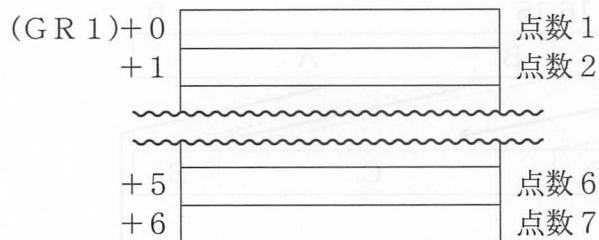
```

- 5 アセンブラー言語CASL IIが使用できる計算機COMET IIがある。次のプログラムの説明及びプログラムを読んで、プログラム中の①～⑤に入れるべき適切な語を解答群から選び、記号で答えなさい。

[プログラムの説明]

7人の採点者のつけた点数を受け取り、最高点一つと最低点一つを除いた五つの点数の合計をGR 0に設定し呼び出しプログラムに戻す、副プログラムGRADである。

(1) 七つの点数は、図のように連続する七つの語に入れられ、その先頭アドレスがGR 1に格納されて呼び出しプログラムから渡される。



(2) 点数は0点以上、100点以下の整数とする。

(3) 呼び出しプログラムに戻るとき、汎用レジスタGR 1～GR 6の内容を元に戻す。

[プログラム]

0 1	GRAD	START
0 2		R PUSH
0 3		LAD GR 2, 1
0 4		LD [①], 0, GR 1 ; 最初の点数をセット
0 5		LD GR 3, GR 0
0 6		LD GR 4, GR 0
0 7		LD GR 5, GR 0
0 8		LAD GR 6, 6 ; ループカウンタ
0 9	LOOP	LAD [②], 1, GR 1
1 0		LD GR 0, 0, GR 1
1 1		C PA GR 0, [③]
1 2		J PL NOTMIN
1 3		LD GR 3, GR 0
1 4		J UMP NOTMAX
1 5	NOTMIN	C PA GR 0, GR 4
1 6		J MI NOTMAX
1 7		LD GR 4, GR 0
1 8	NOTMAX	ADD A GR 5, GR 0
1 9		SUB A [④], GR 2
2 0		J NZ LOOP
2 1		LD GR 0, GR 5
2 2		SUB A GR 0, GR 3
2 3		SUB A GR 0, [⑤]
2 4		R POP [⑥]
2 5		RET
2 6		END

解答群

ア. GR 0

イ. GR 1

ウ. GR 3

エ. GR 4

オ. GR 6

社団法人 全国工業高等学校長協会
 平成22年度前期 第44回1級情報技術検定
試験問題〔I〕標準解答

1

5点×5
計25点

問1		問2	問3	問4
①	②	③	④	⑤
1101.1	0.1B	8	30	-32 768

2

問1
2点×5

問2
2点×4

問3
7点
問4

5点×2

計35点

問1				
①	②	③	④	⑤
0	0	0	1	1
問2				
リセット	クロックパルス	CK	Q_0	Q_1
			1	1
			0	0
			1	1
			0	0
			1	1
			0	0
			1	1
			0	0
			1	1
			0	0
問3		問4		
50	GB	①	②	
		0.72	0.98	

3

2点×10
計20点

問1	①	②	③	④	⑤
	ウ	才	ア	イ	工
問2	①	②	③	④	⑤
	イ	ア	才	ウ	工

4

2点×10
計20点

問1	①	②	③	④	⑤
	キ	サ	コ	才	工
問2	①	②	③	④	⑤
	オ	カ	ウ	イ	工

社団法人 全国工業高等学校校長協会
平成22年度前期 第44回 1級情報技術検定
試験問題〔II〕標準解答

1 4点×5 計20点

①	②	③	④	⑤
n - 1	n	d (i)	d (i)	temp

2 4点×5 計20点

①	②	③	④	⑤
0	1	2	SUM+L→SUM	L→J

3 4点×5 計20点

①	②	③	④	⑤
キ	ア	力	才	ケ

C選択・CASL II選択用

選択する言語を○で囲みなさい。

4 5点×4 計20点

①	②	③	④
C k	2	j	a [j]
CASL II GR 0 , GR 1	GR 0 , GR 2	LB 6	LB 5

5 4点×5 計20点

①	②	③	④	⑤
C ア	オ	ク	ウ	工
CASL II イ	ア	ア	ウ	工

注 標準解答以外でも、論理的に正しいものは正解とする。
ただし、無駄な繰り返しや意味のない代入は行われていないこと。

社団法人 全国工業高等学校長協会
 平成22年度後期 第45回 1級情報技術検定
試験問題〔I〕標準解答

1 5点×5 合計25点

問 1		問 2	問 3	問 4
①	②	③	④	⑤
6866	6B4.8	6	14	D6DA

2 問1 2点×5 小計10点 問2 2点×4 小計8点 問3 5点×2 小計10点 問4 7点 合計35点

問 1										
①	②	③	④	⑤						
1	0	ア	ア	ウ						
問 2										
計数パルス	1	2	3	4	5	6	7	8	9	10
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Q_1	1	1	0	1	0	1	0	1	0	1
Q_2	1	0	1	0	1	0	1	0	1	0
Q_3	1	0	0	1	0	1	0	1	0	1
Q_4	1	0	0	0	1	0	1	0	1	0

3 2点×10 合計20点

	①	②	③	④	⑤
問 1	キ	ウ	ア	工	イ
問 2	①	②	③	④	⑤
	才	力	イ	工	ア

4 2点×10 合計20点

①	②	③	④	⑤
才	コ	キ	ク	ア
⑥	⑦	⑧	⑨	⑩
ケ	力	ウ	シ	サ

社団法人 全国工業高等学校校長協会
平成22年度後期 第45回 1級情報技術検定
試験問題〔II〕標準解答

1 5点×4 合計20点

①	②	③	④
low	high	low	high

2 4点×5 合計20点

①	②	③	④	⑤
I	SUM + J → SUM	=	J	J + 1 → J

3 4点×5 合計20点

①	②	③	④	⑤
s + d	k + 2	EPS	MAX (200でも可)	s

C選択・CASL II選択用

選択する言語を○で囲みなさい。

4 5点×4 合計20点

①	②	③	④
C m[i]	t / m[i]	t% m[i] または t - n[i]*m[i] または t - t / m[i]*m[i]	0
CASL II GR 5	GR 4	GR 0	SAVE

5 4点×5 合計20点

①	②	③	④	⑤
C キ	エ	コ	ア	オ
CASL II ア	イ	ウ	オ	エ

注 標準解答以外でも、論理的に正しいものは正解とする。
ただし、無駄な繰り返しや意味のない代入は行なわれていないこと。

1級 情報技術検定試験 標準問題集 定価 1,800円 (消費税込)	平成23年4月1日 発行 編 集 社団法人 全国工業高等学校長協会 発行所 同 上 〒102-0072 東京都千代田区飯田橋2-8-1 電 話 03-3261-1500(代) 振 替 00160-4-96148	この本は発行所の許可なく ては、複製し またはその内 容を転載して はならない。

A) 予約語 JIS X3010 : 2003(ISO/IEC 9899 : 1999)

auto	enum	restrict	unsigned
break	extern	return	void
case	float	short	volatile
char	for	signed	while
const	goto	sizeof	_Bool
continue	if	static	_Complex
default	inline	struct	_Imaginary
do	int	switch	
double	long	typedef	
else	register	union	

B) 8ビット文字コード表 ANK (JIS X0201)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p		SP	-	タ	ミ			
1	SOH	DC1	!	1	A	Q	a	q	.	。	ア	チ	ム			
2	STX	DC2	"	2	B	R	b	r		「	イ	ツ	メ			
3	ETX	DC3	#	3	C	S	c	s		」	ウ	テ	モ			
4	EOT	DC4	\$	4	D	T	d	t		、	エ	ト	ヤ			
5	ENQ	NAK	%	5	E	U	e	u		・	オ	ナ	ユ			
6	ACK	SYN	&	6	F	V	f	v		ヲ	カ	ニ	ヨ			
7	BEL	ETB	,	7	G	W	g	w		ア	キ	ヌ	ラ			
8	BS	CAN	(8	H	X	h	x		イ	ク	ネ	リ			
9	HT	EM)	9	I	Y	i	y		ウ	ケ	ノ	ル			
A	LF	SUB	*	:	J	Z	j	z		エ	コ	ハ	レ			
B	VT	ESC	+	;	K	[k	{		オ	サ	ヒ	ロ			
C	FF	FS	,	<	L	¥	l			ヤ	シ	フ	ワ			
D	CR	GS	-	=	M]	m	}		ュ	ス	ヘ	ン			
E	SO	RS	.	>	N	^	n	~		ヨ	セ	ホ	・			
F	SI	US	/	?	O	_	o	DEL		ツ	ソ	マ	。			

C) エスケープシーケンス

エスケープシーケンス	意味
¥¥	バックスラッシュ(¥)
¥'	一重引用符(')
¥"	二重引用符(")
¥a	端末ベル(BEL)
¥b	バックスペース(BS)
¥f	フォームフィード(FF)
¥n	行送り(LF)
¥r	復帰(CR)
¥t	水平タブ(HT)
¥v	水平タブ(VT)
¥ooo	8進数値 ooo を持つ文字
¥xhh	16進数値 hh を持つ文字

D)演算子の優先順位

順位	種類	演算子	結合
1	関数,添字,構造体,後置増分減分	() [] . -> ++ --	→
2	前置増分減分,単項式	++ -- ! ~ + - * & sizeof	←
3	キャスト	(型)	←
4	算術乗除余	* / %	→
5	算術加減	+ -	→
6	ビットシフト	<< >>	→
7	関係比較	< <= > >=	→
8	関係等値	== !=	→
9	ビットAND	&	→
10	ビットXOR	^	→
11	ビットOR		→
12	論理AND	&&	→
13	論理OR		→
14	条件	? :	←
15	代入	= += -= *= /= %= &= ^= = <<= >>=	←
16	順次	,	→

E)標準ライブラリ(抜粋)

ヘッダファイル	主な関数
<cctype.h>	文字種別の分類、および大文字・小文字の変換を行う関数 isalnum isalpha isblank iscntrl isdigit isgraph islower isprint ispunct isspace isupper isxdigit tolower toupper など
<math.h>	数学的な演算を行うための関数、および関連するマクロ・関数の宣言定義 acos asin atan cos exp fabs fmod log log10 modf pow sin sqrt tan など
<stdio.h>	ストリームおよびファイルの操作に関する型・マクロ・関数の宣言定義 FILE fclose fgetc fgets fopen fprintf fputc fputs fread fscanf fseek getc gets perror printf putc puts remove rename scanf など
<stdlib.h>	一般ユーティリティに関する型・マクロ・関数の宣言定義 RAND_MAX abort abs atof atoi atol bsearch div exit free getenv labs ldiv malloc qsort rand srand strtod strtol strtoul system など
<string.h>	文字列操作に関する型・マクロ・関数の宣言定義 strchr strcat strcmp strcpy strlen strstr strtok など
<time.h>	日付等を扱うための型・マクロ・関数の宣言定義 time など

ISBN978-4-904677-05-6

C3055 ¥1714E



9784904677056

定価(本体1714円+税)

1923055017146

