

2021年度 A O入試 1次選考問題

情報理工学部

情報科目

注意事項

1. 試験開始の合図があるまで、問題用紙を開いてはいけません。
2. 解答はすべて、所定の解答用紙に記入してください。
3. 解答用紙に受験番号と氏名（フリガナ）を記入してください。
4. 解答時間は60分です。問題は12ページあります。
5. 問題用紙・解答用紙および計算用紙はすべて回収します。一切持ち帰ってはいけません。

〔 I 〕 次の説明文中の空欄（ア）～（ク）に該当する用語を、下記用語選択肢に記したも
のから選択し、A～F の記号で答えよ。

高度に情報化された社会では、世の中の様々な物事や情報、知識が〔ア〕として
表現され、〔イ〕を含むコンピュータ・ネットワークを介して通信される。そこ
では、スマートフォンやPCなどの人が利用する端末に限らず、田畑に設置された気温・湿度
を計測する〔ウ〕のデータや、街中やビルに設置された〔エ〕の画像、
人工衛星からの気象関連データなど、人が直接操作しないコンピュータ機器からの情報も
多く扱われる。光ケーブルでの接続を含む〔オ〕は、高速に通信できることも多い
が、配線できる場所や接続する機器の個数が限られるほか、移動しない機器や装置に利用さ
れることが一般的である。一方で、〔カ〕は、電波の利用周波数帯や通信方式によ
って通信速度や接続数は様々であるものの、電波が有効に届く範囲で柔軟に利用でき、配線
を気にせず移動しながらでも利用できる側面がある。日本国内では2020年3月からは
〔キ〕のサービスが開始され、大都市圏を中心に徐々に利用可能エリアが広がって
いる。その特徴は、それまでの携帯電話の通信規格に比べ、高速大容量、〔ク〕、
多数同時接続である。一般人の〔イ〕利用や動画視聴だけでなく、IoT やスマート
シティなどで利用する通信技術として様々な応用が期待されている。

用語選択肢（A～M）

- | | | | |
|------------|---------------|-----------------|--------|
| A. センサ | B. 無線ネットワーク接続 | C. インターネット | D. カメラ |
| E. アナログデータ | F. 有線ネットワーク接続 | G. 第4世代移動通信システム | |
| H. 低信頼高遅延 | I. マイクロフォン | J. 第5世代移動通信システム | |
| K. ビデオゲーム | L. デジタルデータ | M. 高信頼低遅延 | |

〔Ⅱ〕 とあるネットワーク通信にて、次のメッセージ（バイト列）を受信した。受信したメッセージはすべて ASCII コードで表現された文字コードである。下記の ASCII コード表を参考に、メッセージの内容を文字列として書き出せ。

受信メッセージ（すべて 16 進数）：

48 65 6C 6C 6F 2C 57 6F 72 6C 64 21 0A 4B 53 55

ASCII コード表

上位 3ビット 下位 4ビット	0_	1_	2_	3_	4_	5_	6_	7_
_0	NUL	DLE	SP	0	@	P	`	p
_1	SOH	DC1	!	1	A	Q	a	q
_2	STX	DC2	"	2	B	R	b	r
_3	ETX	DC3	#	3	C	S	c	s
_4	EOT	DC4	\$	4	D	T	d	t
_5	ENQ	NAC	%	5	E	U	e	u
_6	ACK	SYN	&	6	F	V	f	v
_7	BEL	ETB	'	7	G	W	g	w
_8	BS	CAN	(8	H	X	h	x
_9	HT	EM)	9	I	Y	i	y
_A	LF/NL	SUB	*	:	J	Z	j	z
_B	VT	ESC	+	;	K	[k	{
_C	FF	FS	,	<	L	¥	l	
_D	CR	GS	-	=	M]	m	}
_E	SO	RS	.	>	N	^	n	~
_F	SI	US	/	?	O	_	o	DEL

BS : バックスペース LF/NL : 改行 ESC : エスケープ
 CR : キャリッジリターン SP : 空白 DEL : 削除

〔Ⅲ〕 次の文章を読んで、設問(A)から設問(D)に答えよ。

半導体メモリは値を記憶する素子である。一つのメモリセルは複数の半導体素子から構成され、一つのメモリセルは1ビットまたは複数ビットの値を保持する。ただし以下本問では、一つのメモリセルは1ビットの値を保持できるものとする。

設問 (A)

一般的に、1ビットをいくつかまとめて1バイトとして扱う。1バイトは何ビットか答えよ。

設問 (B)

プロセッサ(CPU)は、一般的に1ワード単位でデータの読み書きを行う。また1ワードのビット数はコンピュータが一度に処理できるビット数のことであり、コンピュータによって異なる。現代のコンピュータでは、メモリには1バイトごとにデータが格納され、格納される場所それぞれにアドレス(番地)が付与されている。アドレスによってデータの格納場所を指示できる。

16bitコンピュータ		32bitコンピュータ	
アドレス	データ	アドレス	データ
0000	07AD	00000000	000007AD
0004	1ABF	00000008	00001ABF
0008	2020	00000010	20202020
000B	1958	00000018	00001958
⋮		⋮	
FFF8	0123	FFFFFFFF0	01230123
FFFB	ABCD	FFFFFFF	0000ABCD

図 III-1 アドレス表記とそのデータの例

以下の2つの問いに答えよ。

(問1) 1ワードのビット数によって、CPUは表現できるアドレス数が異なる。1ワード16ビットのコンピュータは、何種類のアドレスを表現できるだろうか。以下の(a)から(d)の選択肢から一つ選べ。

- (a) $16 = 2^4$
- (b) $256 = 2^8$
- (c) $4,096 = 2^{12}$
- (d) $65,536 = 2^{16}$

(問題〔Ⅲ〕の設問のつづき)

(問2) 1ワード32ビットのコンピュータでは、最大に扱えるメモリ容量は何バイトになるか、以下の(a)から(d)の選択肢から一つ選べ。

- (a) $65,536 = 2^{16}$
- (b) $1,073,741,824 = 2^{30}$
- (c) $4,294,967,296 = 2^{32}$
- (d) $17,179,869,184 = 2^{34}$

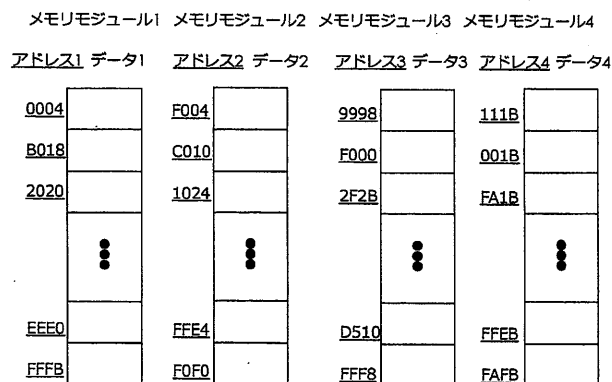
設問 (C)

現在のコンピュータは一つのメモリモジュールに対して、一箇所のアドレスのみ読み書きすることができる。一方、複数のメモリモジュールがある場合、個々のメモリモジュールごとに同時に読み書きすることができるため、複数のアドレスからデータを読み出す時間を短縮することができる。

連続したアドレスから大量のデータを高速に読み書きしたいニーズがある。例えば、動画を再生する際に、連続したアドレスに格納されているデータを読み出す。このようなニーズに対する解決策の一つは、複数のメモリモジュールを用い、個々のメモリモジュールに対して異なるアドレスを割り当て、かつ同時にアクセスを可能とする仕組みである。この仕組みはメモリコントローラを用いて実現されている。

この仕組みを用いるにあたって、複数のメモリモジュールに対して、アドレスをどのように割り振ればより高速にメモリを読み書きできるかについて考える。

図Ⅲ-2に4つのメモリモジュールの例を示す。図Ⅲ-2のメモリモジュールでは、メモリモジュール1から順に小さな値のアドレスを割り振っている。図2のようにアドレスを割り振ることで、アドレス0004とアドレスF004へ同時にアクセスすることができる。

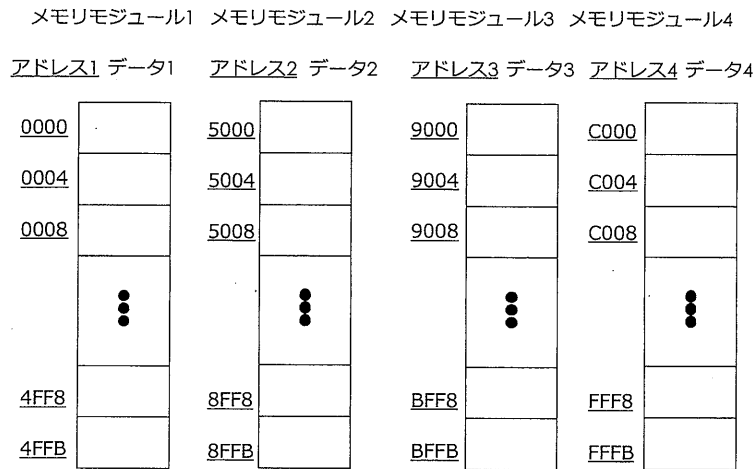


図Ⅲ-2 4つのメモリモジュール

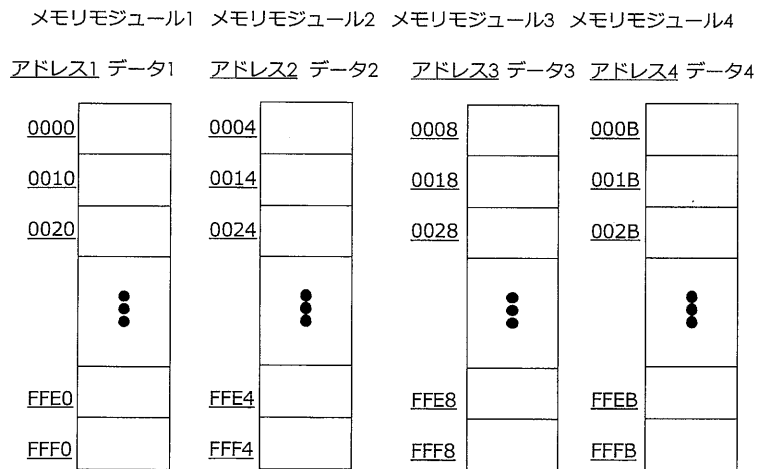
(問題 [III] の設問のつづき)

図III-2のようにメモリモジュールを4つ用いた場合、どのようにアドレスを割り振れば、連続したアドレスにある多量のデータを高速に読み書きできるだろうか。以下の(a)から(d)の選択肢から一つ選び、また選択した理由を簡潔に述べよ。

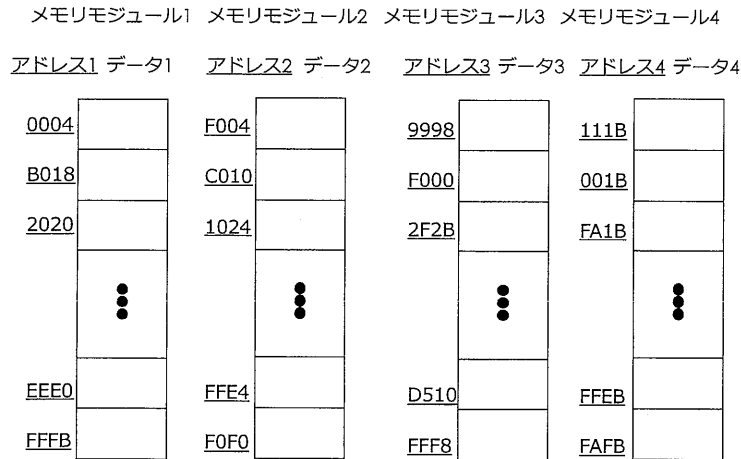
(a)



(b)



(c)



(d) その他

設問 (D)

SSD(Solid State Drive)やUSBメモリで利用されているフラッシュメモリは高速に読み書きできるメモリである。フラッシュメモリは複数のメモリセルから構成され、一つのメモリセルは複数の半導体素子から構成されている。またこの構成のため、フラッシュメモリは同一のメモリセルに対する書き込み可能回数が制限されているという性質がある。この性質により、フラッシュメモリは特定のメモリセルに書き込みが集中した場合、そのメモリセルはいずれ正しい値を保持できなくなる。正しい値を保持できないと判断されたメモリセルは利用されなくなり、ユーザが利用できるメモリサイズは小さくなる。

フラッシュメモリのような同一のメモリセルに対する書き込み回数に制限がある性質を持つメモリに対して、メモリの容量を保つために、メモリコントローラがその読み込み書き込み処理を制御している。

メモリコントローラが、メモリの容量を保つために、どのような読み込み書き込み処理の制御を行えば、メモリの長寿命化につながるかを検討して答えよ。

〔IV〕 以下の文章を読んで、設問 (A) ~ (C) に答えよ。

以下の手順で暗号化/復号化する転置式暗号を考える。

暗号化は以下の 3 つの手順で行われる。

- 1) 図 IV-1 の様に与えられた平文 (この場合は“ABC DEF GHIJ.” が与えられているとする) 中のスペースを“@”、ピリオドを“\$”に置換し、三行四列の二次元配列へ行方向に入れる。

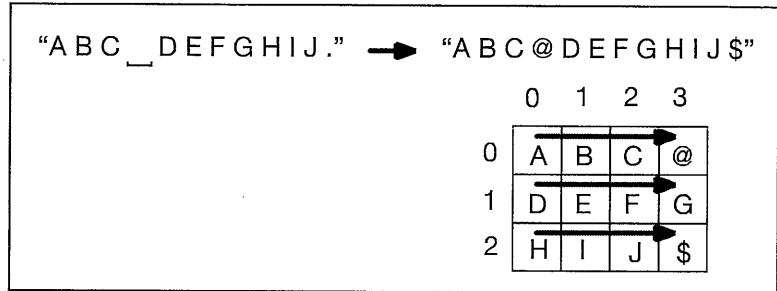


図 IV-1 平文の置換と 2 次元配列への挿入

- 2) 図 IV-2 のように図 IV-1 の配列を左に 180° 回転する。

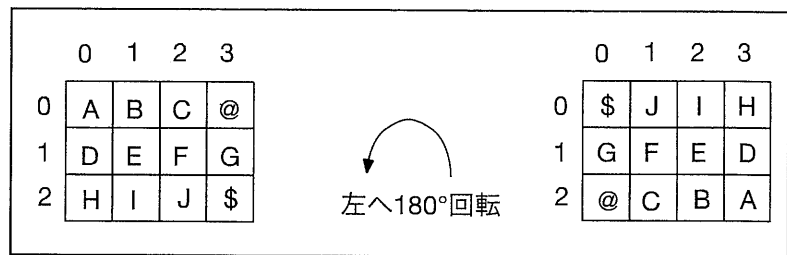


図 IV-2 二次元配列の回転

- 3) 図 IV-3 のように二次元配列を列方向に読み出し、“@”をスペース、“\$”をピリオドに置換し暗号文を得る。

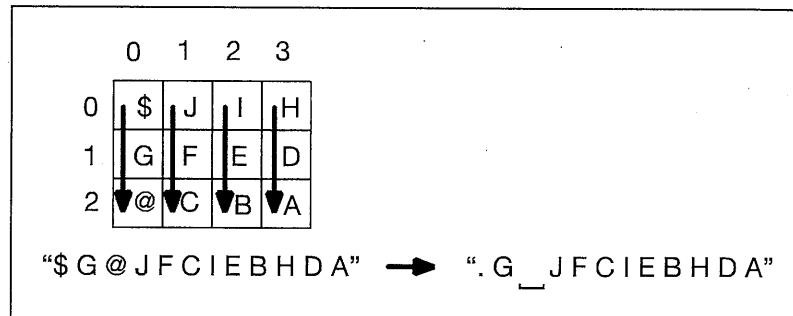


図 IV-3 二次元配列からの読み出しと暗号文の取得

また復号化は、以下の 3 つの手順で行われる。

- 1) 与えられた暗号文を図 IV-3 のように二次元配列へ列方向に挿入する。
- 2) 図 IV-2 のように二次元配列を左に 180° 回転させる。
- 3) 図 IV-1 のように二次元配列を行方向に読み出し、文字列中の「@」をスペース、「\$」をピリオドに置き換える。

(問題 IV の設問のつづき)

設問 (A)

以下の平文を暗号化せよ。

“Hello _ World.”

設問 (B)

以下の暗号文を復号化せよ。

“\$s@eermw@oau”

設問 (C)

図 IV-4 は暗号化を行うプログラムである。

図 IV-4 のプログラム中の空欄(ア)～(ウ)を答えよ。ただし、配列 letters には与えられた文字列中の「@」と「\$」が置換された文字列が入っているものとする。

```
letters[3][4] /* 入力された文字列を格納 */
result[3][4] /* 暗号化された文字列を格納 */

/* 配列 letters を左に 180° 回転した内容を配列 result に代入する */

for (i=0; (ア) )
  for (j=0; (イ) )
    (ウ)
  end
end

/* 暗号化した文字列を出力 */
for (j=0; j < 4; j = j+1)
  for (i=0; i < 3; i = i+1)
    print(result[i][j])
  end
end
```

図 IV-4 暗号化を行うプログラム

〔V〕以下の文章を読んで、設問 (A) ～ 設問 (D) に答えよ。ただし、この問〔V〕では、問題冊子の最後に示す記法を使ってプログラムを記述する。

正整数が 100 個格納された配列 D が与えられている。配列 D は整列されていないが、各値は 150 より小さいことがわかっている。どの値が配列 D に何回含まれているかは次に示すプログラム 1 を実行するとわかる。ただし、配列 C の大きさは 150 であると仮定し、i、j、n、m は変数であるとする。

設問 (A)

プログラム 1 が、配列 D に含まれる各値とその個数を表示するように、空欄 1 を埋めよ。

設問 (B)

中央値とは、与えられたデータを大きさの順に小さい方から並べたときに中央にある値である。ここでは配列 D の値を整列させたときに 50 番目 (先頭は 0 番目とする) にある数を中央値とする。

設問 (A) で配列 C に格納した値を利用すると、配列 D の値を整列させることなく、配列 C の値だけから中央値を得ることができる。プログラム 1 の空欄 2、3 を埋めて、変数 m に配列 D の中央値が得られるようにせよ。

なお、プログラム内にすでにある変数のほか、必要に応じて変数 s、t、u を使ってもよい。

```
for (i = 1; i < 150; i = i + 1)
    C[i] = 0
end

for (j = 0; j < 100; j = j + 1)
    n = D[j]
    【空欄 1】
end

for (i = 1; i < 150; i = i + 1)
    if C[i] > 0
        print(i, "は", C[i], "個")
    end
end

s = 0
for (i = 1; i < 150; i = i + 1)
    if C[i] > 0
        s = 【空欄 2】
        if 【空欄 3】
            m = i
            break
        end
    end
end

print("中央値は", m)
```

プログラム 1

(問 [V] の設問のつづき)

正整数のデータをファイルから読み込むプログラムを考える。データの個数は100個だが、最大値はあらかじめ分からないとする。このような場合、前ページで考えた方法で同じ値が何回現れたかを調べようとしても、値ごとの出現回数を保持する配列 C の大きさを決めることができない。

そこで、右のようなプログラムを作成した。このプログラムでは2つの配列を使って、入力された値とその出現回数を記録する。A と C は大きさ200の配列であり、i、j、k、data は変数である。演算子 % は剰余算を、識別子 true は常に真である条件を表す。また、関数 input() を呼び出すことによって、ファイルから整数値を1つ入力できるとする。

```
for (i = 0; i < 200; i = i + 1)
    A[i] = 0
    C[i] = 0
end
for (j = 0; j < 100; j = j + 1)
    data = input()
    k = data % 200
    while true
        if A[k] == 0
            A[k] = data
            C[k] = 1
            break
        end
        if A[k] == data
            C[k] = C[k] + 1
            break
        end
        k = (k + 3) % 200
    end
end
```

プログラム 2

設問 (C)

ファイルから次のような値の列が読み込まれたとする。

18、632、10032、15、18、801215、15、10032、...

最初の2つの値が読み込まれた時点で、配列 A および C で内容が0ではない箇所の内容は表1のように表すことができる。

表2は、さらにもう1つの値を読み込んだ時点の内容である。

その後、8つめまでのデータを読み込んだ時点での配列 A および C の内容を同様に示せ。

表 1

添字	A	C
18	18	1
32	632	1

表 2

添字	A	C
18	18	1
32	632	1
35	10032	1

設問 (D)

プログラム2ですべての入力データを処理した後、ある値がデータに含まれていたかを調べたい。変数 X の値がデータに含まれていれば「YES」、含まれていなければ「NO」を表示するプログラムを作成せよ。必要に応じて変数 i、j、s、t を使用してもよい。

プログラムの記法の説明

問 [IV] [V] でプログラムの記述に用いる記法について説明する。

文

変数 = 式	変数に式の値を代入する。以下の説明ではこの文を「代入文」と呼ぶ。
for (代入文 1; 条件式; 代入文 2) ... end	まず、代入文 1 を実行し、条件式を評価する。条件式が偽であれば何もしない。条件式が真のとき、end までの命令を実行し、次に代入文 2 を実行する。その後、条件式が真である間、end までの命令と代入文 2 を実行する。
while 条件式 ... end	条件式が真である間、end までの命令を実行する。
if 条件式 ... else ... end	条件式が真のとき、else までの命令を実行し、偽のとき、else から end までの命令を実行する。else を記述しない場合、条件式が真のときに end までの命令を実行し、条件式が偽のときは何もしない。
break	for 文、または while 文（これらをループ文と呼ぶ）の内部でのみ利用できる。実行すると繰り返しの処理を打ち切り、ループ文の次の文の実行に移る。ループ文の中でループ文が使われている（ネストしている）時は、一番内側のループ文の処理だけが打ち切られる。
print(式)	式の値を表示する。

配列

配列名[式] 含まれる要素の数があらかじめ決められた配列を利用できる。配列 a の要素数が N (N は正の整数) のとき、配列 a は 0 番目から (N-1) 番目までの要素を持ち、i 番目の要素は a[i] という記法で表現する。

比較演算子

A == B	A と B の値が等しい。
A != B	A と B の値が等しくない。
A <= B	A の値が B の値以下である。
A < B	A の値が B の値より小さい。

A >= B Aの値がBの値以上である。
A > B Aの値がBの値より大きい。

算術演算子

+ 加算（足し算）を行う。
- 減算（引き算）を行う。
* 乗算（掛け算）を行う。
/ 除算（割り算）を行う。ただし、結果は実数で表される。
% 剰余算を行う。剰余算は割り算の余りを求める。例えば 7%3 は 1 となる。

複数の算術演算子が混在した式では * と / と % の計算が + や - よりも先に行われる。
算術演算子と比較演算子が混在した式では、算術演算子が先に計算される。
また、式の中で () を使い、計算の順序を示すことができる。

コメント

プログラム中に現れる /* と */ に囲まれた文字列はコメントとみなし実行されないものとする。

```
s = 0 /* 変数sに0を代入する */  
i = 1 /* 変数iに1を代入する */  
while s <= 100  
    s = s + i  
    print(s)  
    i = i + 1  
  
for (i = 0; i < n; i = i + 1)  
    if a[i] >= 0  
        b[i] = a[i]  
    else  
        b[i] = 0  
    end
```

プログラムの例

- (1) $1 + 2 + 3 + \dots$ と加算を繰り返して、その値を表示する。合計が 100 を超えたら終わる。
- (2) 与えられた要素数 n の配列 a の内容を、同じ大きさの配列 b にコピーする。ただし、配列 a の要素で負の数があれば、 b には代わりに 0 を代入する。