北海道大学　大学院情報科学院

情報科学専攻　修士課程

情報理工学コース

専門科目１

１０：００～１２：００

受験上の注意

● 本冊子内の５問，問１（基礎数学），問２（情報数学），問３（確率・統計），問４（コンピュータ基礎工学），および問５（プログラミング）から３問を選択し解答すること．

●すべての解答用紙に，受験番号，選択した問題番号(例えば，問３など)を記入すること．

● 選択問題チェック票に受験番号および，選択した科目に印を記入すること．

●解答用紙は３枚である．この他に下書き用の草案紙３枚を配付する．

●解答は，問題ごとに別々の解答用紙に記入すること(裏面を使用してもよい．解答用紙を破損したりした場合には試験監督員に申し出ること)．

●問題冊子，草案紙は持ち帰り，選択問題チェック票とすべての解答用紙を提出すること．

●机の上に置いてよいものは，筆記用具（黒鉛筆，消しゴム，鉛筆削り），時計，および特に指示があったもののみである．時計は計時機能のみを使用し，アラームの使用を禁ずる．携帯電話，スマートフォン，タブレット，コンピュータ等は電源を切ってかばんの中にしまうこと．電卓，電子辞書などは使用を禁ずる．

問 1. 基礎数学

［ 1 ］ 行列に関する以下の問いに答えよ．

(1) 3 次元空間内の異なる非零な 3 点 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y} \in \mathbb{R}^3$ に対して，方程式 $a\mathbf{x}_1 + b\mathbf{x}_2 = \mathbf{y}$ が解 (a, b) を持つとき，この 3 点は一次（線形）従属であることを証明せよ．

(2) 各行各列に一つだけ 1 があり他の要素がすべてゼロである n 次正方行列 A は，変換 $A : \mathbb{R}^n \to \mathbb{R}^n$ としてどのようなものか，点 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$（ここで，$T$ は転置）を変換して説明せよ．

(3) 前問 (2) の性質を持つ任意の n 次正方行列 A は，次のような n 次正方行列 P_{ij} の積として分解できることを示せ．ここで，P_{ij} は，i 行では j 列にだけ 1，j 行では i 列にだけ 1，それら以外の行は対角要素だけ 1，で残りはゼロである n 次正方行列である．

(4) 一方が他方の定数倍ではない 3 次元空間内の 2 点 $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ を考え，これらを横に並べてできる 3×2 の行列 $A = (\mathbf{x}_1\ \mathbf{x}_2)$ を作る．このとき，$A^T A$ は正則であることを証明せよ．ここで，A^T は A の転置を表す．

(5) 前問 (4) の $3{\times}2$ の行列 $A = (\mathbf{x}_1\ \mathbf{x}_2)$ において，$3{\times}3$ 行列 $P = A(A^T A)^{-1} A^T$ を作る．このとき，P は，任意の 3 次元空間内の点を，$\mathbf{x}_1$ と $\mathbf{x}_2$ で張られる平面へ垂直に射影（直交射影）することを示せ．

［ 2 ］ 級数や関数などに関する以下の問いに答えよ．ただし，実数の連続性 (完備性) は仮定してよい．他に定理などを使う場合は名前を挙げること．

(1) 関数 $\log x$ の $x = 1$ における 3 次までのテイラー展開を求めよ．ただし，$x \in (0, 2)$ とする．ここで，log は自然対数である．

(2) 級数 $s_n = a_1 + a_2 + \cdots + a_n, n = 1, 2, \dots$ に関する次の命題に関して，逆，裏，対偶を述べ，次の命題を含む 4 つの命題の真偽を述べよ．さらに，偽の場合は「反例」を挙げよ．

$$a_n \text{ は } 0 \text{ に収束する} \Rightarrow s_n \text{ は有限確定値に収束する}$$

(3) 人の集合 P と述語 LOVES(x, y)（意味は「x は y を好きである」）に対して，次の二つの述語論理式

$$A : \forall x \in P\, \exists y \in P\ \mathrm{LOVES}(x, y), \qquad B : \exists y \in P\, \forall x \in P\ \mathrm{LOVES}(x, y)$$

の関係を述べよ（等価，含意，など）．

(4) 連続実関数 $f(x)$ において $\max f(x)$ が存在せず $\sup f(x)$ が存在するような $f(x)$ と x の区間の例を示せ．

(5) 閉区間 $[a, b]$ で微分可能で $f'(x) > 0$ である実数値関数 $f(x)$ は $[a, b]$ で単調増加であることを示せ．

問 2. 情報数学

[1] 以下の命題論理式それぞれについて，その積和標準形 (DNF; 選言標準形) および和積標準形 (CNF; 連言標準形) を求めよ．ただし，$\vee$ は論理和，$\wedge$ は論理積，$\neg$ は否定，$\to$ は含意を表す．なお，どの論理式が求めた DNF や CNF であるかを明示すること．

(1) $((x \to \neg y) \to (\neg y \to z)) \wedge (x \vee \neg y)$

(2) $(x \to y \wedge z) \wedge (y \wedge z \to x)$

[2] 整数全体の集合を $\mathbb{Z}$ とし，$\mathbb{Z}$ 上での二項関係 $x \simeq y$ は，$x = y + 3k$ を満たすような $k \in \mathbb{Z}$ が存在することを表すとする．この二項関係 $\simeq$ について，以下の問いに答えよ．

(1) 二項関係 $x \simeq y$ が同値関係であることを証明せよ．

(2) 同値関係 $x \simeq y$ に対して，$\mathbb{Z}$ の $\simeq$ による商集合 (すなわち，すべての同値類の集合) $\mathbb{Z}/\simeq$ を示せ．

[3] 無向グラフ $G = (V, E)$ が $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $|V_1| = m$, $|V_2| = n$, $E = V_1 \times V_2$ を満たす時に，このグラフを完全二部グラフ $K_{m,n}$ とよぶ．ここで，頂点集合を $V = \{v_1, v_2, \ldots, v_{m+n}\}$ とする．この完全二部グラフ $K_{m,n}$ において，以下を満たす頂点の系列 $v_{i_0}, v_{i_1}, \ldots, v_{i_k}$ を長さ k の単純パスとよぶ．ここで，$v_{i_j} \in V$ である．

任意の $0 \le j < k$ に対して $(v_{i_j}, v_{i_{j+1}}) \in E$ が成立し，任意の $0 \le j < j' \le k$ に対して $v_{i_j} \neq v_{i_{j'}}$ が成立する．

なお，2つの系列 $v_{i_0}, v_{i_1}, \ldots, v_{i_{k-1}}, v_{i_k}$ と $v_{i_k}, v_{i_{k-1}}, \ldots, v_{i_1}, v_{i_0}$ は同じ単純パスとみなす．

(1) $K_{3,2}$ での長さ最大の単純パスをすべて図示せよ．

(2) $K_{m,2}$ $(m > 2)$ での長さ最大の単純パスの個数を求めよ．

(3) $K_{m,n}$ $(m > n)$ での長さ最大の単純パスの個数を求めよ．

(4) $K_{n,n}$ での長さ最大の単純パスの個数を求めよ．

問 3. 確率・統計

以下の問いに答えよ．ただし，答えだけでなく導出の過程も分かるように解答すること．

[1] 独立同分布な確率変数 $X_1, X_2, \ldots, X_n$ について，それぞれの分散が σ^2 であるとする．

(1) 確率変数 $S = \sum_{i=1}^{n} X_i$ の分散 $\mathrm{Var}(S)$ を求めよ．

(2) 確率変数 $T = X_1/n$ の分散 $\mathrm{Var}(T)$ を求めよ．

[2] 確率変数 X は 0 と 1 のいずれかの値をとり，その確率が $\mathrm{P}(X=1) = p, \mathrm{P}(X=0) = 1-p$ であるとする．ただし，p は $0 \leq p \leq 1$ を満たし，成功確率と呼ばれる．この確率変数 X がしたがう確率分布は成功確率 p のベルヌーイ分布と呼ばれ，$\mathrm{Ber}(p)$ と書かれる．

(1) ベルヌーイ分布 $\mathrm{Ber}(p)$ にしたがう確率変数 X の分散 $\mathrm{Var}(X)$ を求めよ．

(2) 確率変数 $X_1, X_2, \ldots, X_n$ が独立にベルヌーイ分布 $\mathrm{Ber}(p)$ にしたがうとする．このとき，確率変数 $Y = \sum_{i=1}^{n} X_i$ の分散 $\mathrm{Var}(Y)$ を求めよ．

(3) 確率変数 Y の確率関数 $P(Y=k)\,(k \in \mathbb{Z})$ を求めよ．ただし，$\mathbb{Z}$ は整数全体の集合である．

(4) ベルヌーイ分布 $\mathrm{Ber}(p)$ にしたがう独立な確率変数 $X_1, X_2, \ldots, X_n$ について，これを観測値とみなして成功確率 p の推定を行う．成功確率 p の最尤推定量 $\hat{p}$ を求めよ．

(5) 最尤推定量 $\hat{p}$ の分散 $\mathrm{Var}(\hat{p})$ を求めよ．

問4．コンピュータ基礎工学

［1］コンピュータの数値表現に関する以下の問いに答えよ．

(1) 10 進数の 56 を 2 進数で表せ．

(2) 浮動小数点数の算術演算における「丸め誤差」はどのような場合に生ずるか簡潔に述べよ．

［2］カルノー図を用い，以下に示す論理式Qを簡単化せよ．カルノー図は解答用紙に図示すること．

$$Q = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}BCD + ABCD + \bar{A}B\bar{C}D + ABC\bar{D} + A\bar{B}C\bar{D} + AB\bar{C}D$$

［3］計算機アーキテクチャに関する以下の問いに答えよ．

(1) フォン・ノイマン・ボトルネックとはどのようなものか簡潔に述べよ．

(2) フォン・ノイマン・ボトルネックの解消に有効な技術の一つとしてキャッシュメモリがある．キャッシュメモリに関して述べた以下の文中の空欄 I ~ IV に当てはまる最も適切な語句を(a)~(h)から選択し，解答用紙にそれぞれ対応付けて示せ．（例：I-(a)）

「キャッシュメモリはメインメモリと比べてその容量は $\boxed{\quad \mathrm{I} \quad}$ が，そのアクセス速度は $\boxed{\quad \mathrm{II} \quad}$ である．キャッシュメモリを有効に活用するためには，メモリアクセスの $\boxed{\quad \mathrm{III} \quad}$ を高めることが重要である．例えば，大きな配列を走査する場合は，各要素へのアクセスをメモリ上で連続化することで，メモリアクセスの $\boxed{\quad \mathrm{IV} \quad}$ $\boxed{\quad \mathrm{III} \quad}$ を高めることができる．」

(a) 大きい (b) 小さい (c) 高速 (d) 低速 (e) 汎用性 (f) 局所性
(g) 時間的 (h) 空間的

(3) キャッシュメモリのヒット率をh $(0 \leq h \leq 1)$，そのアクセス時間を t_cとし，メインメモリへのアクセス時間をt_mとするとき，メインメモリへの実効アクセス時間t_eをh，t_c，t_mにより表せ．

また，$t_m = 10t_c$である場合のヒット率hと実効アクセス時間t_eの関係を表すグラフの概形（横軸をh，縦軸をt_eとする）を解答用紙に図示せよ．

(4) キャッシュメモリの代表的なデータ格納構造として，ダイレクトマップ方式とセットアソシアティブ方式がある．今，幅 32 ビットのアドレスで管理されるメインメモリに対して，ダイレクトマップ方式を用いたキャッシュメモリを設計したところ，タグのビット数が 16 ビットとなった．キャッシュメモリの容量，キャッシュライン（キャッシュブロック）のサイズを変更せずに，データ格納構造を 4 ウェイのセットアソシアティブ方式に変更した場合，タグのビット数はいくつになるか，その理由とともに述べよ．

次頁へ続く

［4］プロセッサの実行モードである特権モードと非特権モードの違いについて，簡潔に述べよ.

問5．プログラミング

［1］　クレジットカード番号に誤りがないかを確認するために Luhn アルゴリズムが用いられている．このアルゴリズムは入力されたクレジットカード番号（数字の列）が「正しい」か「誤り」かを以下の手順で判定する．

1. A=$A[0]$...$A[N$-$1]$として表現される整数配列 A で N 桁のクレジットカード番号を受け取る．
2. 配列 A の最後の要素（N-1 番目）を 1 番目として，最後の要素から先頭に向かって数えて偶数番目の要素の数字を 2 倍する．ただし，2 倍した数字が 2 桁になった場合には，2 倍した数字の 1 桁目と 2 桁目の数字を足す．
3. 手順 2 で計算した配列に格納された N 桁の数字の総和を求め，変数 S（S は整数）に格納する．
4. 手順 3 で求めた総和 S の 10 進数表現の 1 の位が 0 であれば「正しい」，そうでなければ「誤り」と判定する．

例えば 15 桁のクレジットカード番号「371449635398431」が与えられたとき，

入力されたクレジットカード番号：	3	7	1	4	4	9	6	3	5	3	9	8	4	3	1
最後の要素から偶数番目を 2 倍する：	3	5	1	8	4	9	6	6	5	6	9	7	4	6	1
2 倍した数字が 2 桁になる例：8×2＝16　のため　1+6=7 となる．															
総和：80															

となり，1 の位が 0 のため「正しい」と判定される．

我々は Luhn アルゴリズムを C 言語で実装することとした．ソースコード 1 のア～キを適切に埋めてプログラムを完成させよ．また，下線が引かれている部分には別解が存在する．クとして別解を示せ．

ソースコード 1

```
#include <stdio.h>

int luhn( [   ア   ] ) {
        int i, current_number;
        int index = 0;
        int sum = 0;

        while(number[index] != [   イ   ] ) index++;

        for(i = [  ウ  ] ;i <= index;i++) {
                current_number = [   エ   ] - '0'; // char 型から int 型の
                                                  // 数字に変換
                if(i % [  オ  ] == 0 ) { // 偶数奇数の判定
                        current_number = current_number * 2;
```

次頁へ続く

```
                    if(current_number [  カ  ] 10) {
                            current_number = 1 + current_number % [ キ ]; // ク
                    }
            }
            sum += current_number;
    }

    if(sum % [  キ  ] == 0) { // 総和の 1 の位が 0 かどうかを判定
            return 1;
    } else {
            return 0;
    }
}

int main() {
    char number[] = "371449635398431";

    if(luhn(number) != 0) {
            printf("正しい¥n");
    } else {
            printf("誤り¥n");
    }

    return 0;
}
```

実行結果 1

```
正しい
```

[2]　上記の実装は外部から入力を受け取っておらず，柔軟性に乏しい実装であると指摘を受けた．このため標準入力からクレジットカード番号の桁数と番号を受け取り，動的にメモリを確保する実装に変更した．あわせて luhn 関数内の計算もすべてポインタ変数を使うことにした．ソースコード 2 の A～J を適切に埋めてプログラムを完成させよ．

ソースコード 2

```
#include <stdio.h>
#include <[   A   ]>

int luhn(char *ptr) {
    char *head_ptr = [ B ]; // 受け取った変数の先頭のアドレスを
                            // char *head_ptr で保存する
    int current_number;
```

次頁へ続く

```
        int index = 1;
        int sum = 0;

        while(*ptr != [[1]のイ]) [    C    ];
        ptr--;

        do {
                current_number = [ D ] - '0';

                if(index % [ [1] の オ ] == 0) {
                        current_number = current_number * 2;
                        if(current_number [ [1] の カ ] 10) {
                            current_number = 1 + current_number % [ [1] の キ ];
                        }
                }

                sum += current_number;
                index++;
        } while(ptr-- != [    E    ]);

        if(sum % [ [1] の キ ] == 0) return 1;
        else return 0;
}

int main() {
        char *number;
        int length;

        printf("Length of the number?¥n> ");
        scanf("%d", [    F    ]);
        printf("Card number?¥n> ");

        if((number = ([    G    ]) malloc([      H      ])) == NULL) return 1;
        scanf("%s", [    I    ]);

        if(luhn([    J    ]) != 0) {
                printf("正しい¥n");
        } else {
                printf("誤り¥n");
        }

        free(number);
        return 0;
}
```

次頁へ続く

実行結果 2

```
Length of the number?
> 15 ⏎ // 入力
Card number?
> 371449635398430 ⏎ // 入力
誤り
```