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

閉区間 $I := [-1, 1]$ で定義された，無限回微分可能な実数値関数全体からなる集合 $C^\infty(I)$ は，通常の関数の和と定数倍に関して線形空間 (ベクトル空間) をなす．また，任意の $f(x)$, $g(x) \in C^\infty(I)$ に対して

$$\langle f(x), g(x) \rangle := \int_{-1}^{1} f(x)g(x)dx$$

と定義すると，$\langle f(x), g(x) \rangle$ は $C^\infty(I)$ の内積となる．

自然対数の底を e とし，$\cosh x := (e^x + e^{-x})/2$, $\sinh x := (e^x - e^{-x})/2$ という二つの関数を定義する．更に，I 上で定義された関数の集合 $F(I)$ を

$$F(I) := \{a \cosh x + b \sinh x \mid a, b \in \mathbf{R}\}$$

によって定義する．ここで，$\mathbf{R}$ は実数全体からなる集合を表す．以下の設問に答えよ．

(1) 以下の二つの関数の微分を計算せよ．

$$\text{(i)}\ \cosh x \qquad \text{(ii)}\ \sinh x$$

(2) $F(I)$ が $C^\infty(I)$ の**部分集合**となることを証明せよ．

(3) $F(I)$ が $C^\infty(I)$ の**線形部分空間**となることを証明せよ．

(4) $\cosh x$ と $\sinh x$ が，上で定義した内積に関して直交することを示せ．

(5) 関数の組 $\{\cosh x, \sinh x\}$ が $F(I)$ の基底となることを証明せよ．

(6) 微分が，$F(I)$ から $F(I)$ への**線形写像**となることを証明せよ．

(7) $E : F(I) \to \mathbf{R}^2$ を，関数 $f(x) := a \cosh x + b \sinh x \in F(I)$ から，基底 $\{\cosh x, \sinh x\}$ に対する係数を並べた列ベクトルを生成する写像，すなわち，

$$E(a \cosh x + b \sinh x) := \begin{bmatrix} a \\ b \end{bmatrix}$$

とすると，写像 E は全単射となる．今，$f(x)$, $g(x) \in F(I)$ が $g(x) = \frac{d}{dx} f(x)$ を満たすとき，この関係式と

$$E(g(x)) = A\ E(f(x))$$

が等価となるような 2×2 の実行列 A を求めよ．

(8) 前問で得た行列 A の固有値・固有ベクトルの組を全て求めよ．

(9) $F(I)$ の基底 $\{b_1(x), b_2(x)\}$ のうち，λ_1, λ_2 を何らかの実数とした以下の微分方程式の解となるものを求めよ．また，そのときの λ_1, λ_2 を与えよ．

$$\frac{d}{dx} b_1(x) = \lambda_1 b_1(x), \qquad \frac{d}{dx} b_2(x) = \lambda_2 b_2(x)$$

問２．情報数学

ポイントを使用して景品と交換する．3 種類の景品があり，それぞれ 10 ポイント，5 ポイント，1 ポイントで交換可能である．例えば 10 ポイントを使用する場合の景品の組み合わせ数は，10 ポイントの景品 1 個，5 ポイントの景品 2 個，1 ポイントの景品 10 個，5 ポイントの景品 1 個と 1 ポイントの景品 5 個，の 4 通りである．以下の問いに答えよ．

［1］　20 ポイントを使用する場合の景品の組み合わせ数を求めよ．

［2］　$10n$ ポイント（n は自然数）を使用する場合の景品の組み合わせ数を a_n とする．a_n と a_{n+1} との関係を表す漸化式を理由を添えて示せ．

［3］　［2］の漸化式を解いて a_n を求めよ．

［4］　4 種類の景品があり，それぞれ 50 ポイント，10 ポイント，5 ポイント，1 ポイントで交換可能である．$50n$ ポイント（n は自然数）を使用する場合の景品の組み合わせ数を求めよ．

問３．確率・統計

以下の問に答えよ．ただし，導出の過程も分かるように解答すること．

［１］ １から６までの自然数が各面に記され，振った際に等確率で各面が上面に現われると仮定できるサイコロがある．
いま，参加費としてコインB枚を先に支払い，このサイコロを最大でN回振り，n回目$(1 \leq n \leq N)$に上面に初めて５以上が現れたとき，3^n枚のコインを受け取って終了し，N回すべてで４以下が現れた場合は敢闘賞として１枚のコインを受け取るゲームを考える．このとき，以下の小問に答えよ．

(1) ゲーム運営側が損をしないように，参加費Bを，参加者が受け取るコインの枚数の期待値を下回らない設定にすることを考える．$N = 4$ でのBの最小値と，それを上回る枚数のコインを参加者が受け取る確率を求めよ．

(2) ある参加者から，「これでは参加者側が損をする．たとえば，すべて４以下になるのも珍しいのだから，敢闘賞のコインの枚数をもっと多くすべき．」との提案があった．ゲーム運営側が損をしないように，まず（1）のように敢闘賞をコイン１枚に設定し，参加費Bを，参加者が受け取るコインの枚数の期待値を下回らない設定にしたのち，敢闘賞のコインを増やすことを考える．このときの最大枚数aを，Nおよび実数xを超えない最大の整数を示す記号$[x]$を用いた式で答えよ．

(3) 「５以上が出るまで，何度でもサイコロを振りたい．そのかわり，参加費は，サイコロを振るたびに払うことにし，１回目は１枚，２回目以降は前の回の２倍の枚数を払う．」という参加者が現れた．この申し出は参加者側に有利か不利か，あるいは，どちらにも有利とならないか，理由を付して答えよ．

［２］ 確率変数Xが区間$(0,1)$での連続一様分布に従うとき，以下の小問に答えよ．

(1) Xの累積分布関数，確率密度関数，期待値，分散をそれぞれ答えよ．

(2) $Y = -2\ln(1 - X)$とおく（$\ln(X)$は，Xの自然対数）．
Yの累積分布関数，確率密度関数，期待値，分散をそれぞれ答えよ．

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．

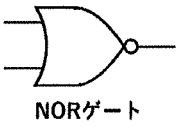
(1) 16 進数の A9 を 2 進数で表せ．

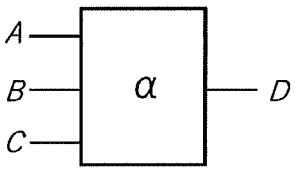
(2) 8 進数の 754 を 2 進数で表せ．

(3) 10 進数の－65 を 8 ビット，2 の補数表現で表せ．

(4) 10 進数の計算 85－12 を 8 ビット，2 の補数表現で計算した場合の計算過程と結果を示せ．

［２］ NOR（否定論理和）ゲートを用いて，論理式 $D = A + B \cdot \bar{C}$ を表す組み合わせ論理回路 α を構成せよ．また，論理回路 α の真理値表を示せ．ここで，＋は論理和，・は論理積，$\bar{}$ は否定を表す．ゲートは何個用いてもよい．


NORゲート



［３］ ノイマン型計算機の動作について述べた以下の文中の空欄（ア）～（オ）に当てはまる最も適切な語句を下段の（a）～（j）から選択し，解答用紙にそれぞれ対応付けて記号で示せ．

CPU が直接実行する機械語プログラムはデータとともに（　ア　）に格納される．CPU は（　イ　）が示す番地に格納されている命令語を（　ア　）から読み出し，その意味を解釈し，逐次実行する．一般に個々の命令語は命令の種類を示す（　ウ　）と命令の対象を示す（　エ　）から構成される．（　エ　）が対象のデータの格納番地である場合には，これを（　オ　）アドレッシングと呼ぶ．

(a)オペランド　(b)即値　(c)直接　(d)間接
(e)インデックス　(f)オペコード　(g)主記憶　(h)スタック
(i)ハッシュ　(j)プログラムカウンタ

次頁へ続く

［4］ 以下のプロセスの一覧表に示される3つのプロセスA，B，Cを単一プロセッサコアで処理する．このとき，以下の条件(1)，(2)，(3)のそれぞれにおいて，下に示す例を参考にスケジューリング結果を図示するとともに，各プロセスのターンアラウンドタイム（プロセスの到着から処理完了までの所要時間）を求めよ．ただし，プロセス切り替えのオーバーヘッドは無視する．

(1) 到着順

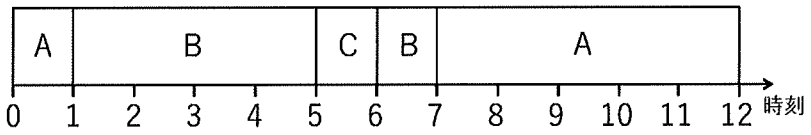
(2) ラウンドロビン（タイムクォンタムを3とする）

(3) ラウンドロビン（タイムクォンタムを2とする）

プロセスの一覧表

プロセス	到着時刻	処理時間
A	0	6
B	1	5
C	5	1

プロセスのスケジューリングの例



問5. プログラミング

［1］ソースコード1は，2つの複素数の積を計算して標準出力に表示するC言語プログラムである．空欄を適切に埋めてプログラムを完成させよ．

ソースコード1

```
#include <stdio.h>

[          (ア)          ];
struct COMPLEX MultiComplex ( struct COMPLEX a, struct COMPLEX b);
void PrintComplex ( struct COMPLEX a );

int main(void)
{
        struct COMPLEX a = { 1.0, 2.0 }, b = { 3.0, 5.0 }, c;

        c = MultiComplex(a, b);
        PrintComplex(c);
        return 0;
}

struct COMPLEX MultiComplex (struct COMPLEX a, struct COMPLEX b ) {
        struct COMPLEX c;
        c.re = a.re * b.re - a.im * b.im;
        c.im = [          (イ)          ];
        return c;
}

void PrintComplex( struct COMPLEX a ) {
        printf("%f%s%fi", a.re, (a.im >= 0.0) ? "+" : "", a.im);
}
```

［2］ソースコード2は，標準入力から読み込んだ自然数に対して，その階乗を計算し標準出力に表示するC言語プログラムである．空欄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>

int factorial(int n);

int main(void)
{
```

次頁へ続く

```
        int a, b;
        scanf("%d", &a);
        if (a < 1) {
                printf("自然数を入力してください¥n");
        }
        else {
                b = factorial(a);
                printf("%d¥n", b);
        }
        return 0;
}

int factorial(int n) {
        if (n == 1) {
                return (ウ);
        }
        else {
                return          (エ)          ;
        }
}
```

［3］ソースコード3は，乱数を用いて近似的に求めた半径1の四分円の面積から，円周率を計算し標準出力に表示するC言語プログラムである．空欄を適切に埋めてプログラムを完成させよ．ここでrand()は0以上RAND_MAX以下の一様整数乱数を返す関数である．

ソースコード3

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
        int i, count;
        int max_iter = 1000000;
        double x, y, pi;
        (オ)          ;
        for (i = 0; i < max_iter; i++) {
                x = (double)rand() / (RAND_MAX + 1.0);
                y = (double)rand() / (RAND_MAX + 1.0);

                if (x * x + y * y < 1) {
                        count++;
                }
```

次頁へ続く

```
        }

        pi = 4.0 * [          (カ)          ];
        printf("%f¥n", pi);
        return 0;
}
```

［4］ソースコード4は，与えた関数funcの定積分を長方形近似で計算し標準出力に表示するC言語プログラムである．空欄を適切に埋めてプログラムを完成させよ．(2箇所の空欄(キ)には同じコードが入る.)

ソースコード4

```
#include <stdio.h>

double func(double x);
double integral([          (キ)          ], double min, double max, int steps);

int main(void)
{
        double min = 0.0;
        double max = 1.0;
        int steps = 1000;
        double s;

        s = integral([     (ク)     ], min, max, steps);
        printf("%f¥n", s);
        return 0;
}

double func(double x) {
        return x * x * x;
}

double integral([          (キ)          ], double min, double max, int steps) {
        int i;
        double x = min;
        double h = (max - min) / steps;
        double sum = 0.0;

        for (i = 0; i < steps; i++) {
                sum += fp(x);
                x += h;
```

次頁へ続く

```
        }
        return (h * sum);
}
```

［5］2001 年 1 月 1 日は月曜日であった．ソースコード 5 は，2001 年の日付を標準入力から読み込み，曜日を標準出力に表示する C 言語プログラムである．日付は，例えば 5 月 3 日は 5/3，10 月 11 日は 10/11 という形式で与える．ただし，存在しない日付に対するエラーチェックは省略している．曜日は英語で出力する．なお，2001 年は「うるう年」(leap year) ではなく，2 月は 28 日間である．空欄を適切に埋めてプログラムを完成させよ．

ソースコード5

```
#include <stdio.h>

int main(void)
{
        int i, month, day;
        int total_days = 0;
        int days_of_month[] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
        [ (ケ) ] day_of_week[] = { "Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday" };

        scanf("%d/%d", &month, &day);


        for (i = 0; i < month-1; i++) {
                total_days += days_of_month[i];
        }

        total_days += day;

        printf("%s¥n", day_of_week[ [ (コ) ] ]);
        return 0;
}
```