

筑波大学 情報学群 情報科学類・情報メディア創成学類

平成 26 年度 学群編入学試験

学力検査問題(専門科目)

[注意事項]

1. 試験開始の合図があるまで、問題の中を見てはいけません。
2. 解答用紙の定められた欄に、学群、学類(併願者は第一志望の学類)、氏名、受験番号を記入すること。
3. この問題冊子は全部で 12 ページ(表紙、白紙を除く)です。
4. 専門科目の選択について、
(ア) 情報科学類と情報メディア創成学類を併願する者は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。ただし、情報メディア創成学類の合否判定においては、数学と情報基礎の解答のみを評価します。
(イ) 情報科学類を単願する者は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。
(ウ) 情報メディア創成学類を単願する者は、問題 1 から問題 4(数学、情報基礎)の計 4 問をすべて答えなさい。
5. 解答用紙は、専門科目で選択した 4 問に対して、各問 1 枚の合計 4 枚を用いること。
6. 解答用紙上部の

 欄に解答する問題番号を記入すること。

問題 1 数学 (1)

- (1) 次の極限值を求めなさい. ただし, $a > 0, b > 0$ とする.

$$\lim_{x \rightarrow 0} \left(\frac{a^x + b^x}{2} \right)^{\frac{1}{x}}$$

- (2) 次の二重積分を求めなさい. ただし, $a > 0$ とする.

$$\iint_D e^{-(x^2+y^2)} dx dy, \quad D = \{(x, y) | x^2 + y^2 \leq a^2, xy \geq 0\}$$

問題2 数学(2)

\mathbf{R}^3 のベクトルを

$$\mathbf{v}_1 = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}, \mathbf{v}_3 = \begin{pmatrix} 3 \\ 1 \\ -4 \end{pmatrix}, \mathbf{v}_4 = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (a, b, c \in \mathbf{R})$$

とする.

- (1) $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ が生成するベクトル空間の基底を一組求めなさい.
- (2) $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ が生成するベクトル空間の次元が3となる a, b, c の条件を求めなさい.
- (3) 行列 $A = (\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_4)$ を用いて, 1 次写像 $f: \mathbf{R}^3 \rightarrow \mathbf{R}^3$ を

$$f(\mathbf{x}) = A\mathbf{x} \quad (\mathbf{x} \in \mathbf{R}^3)$$

によって定める. f の核を求めなさい.

問題3 情報基礎 (1)

素数とは、1と自分自身以外に正の約数を持たない、1でない自然数のことである。素数を求める簡単なアルゴリズムに「エラトステネスのふるい」がある。図1と図2は、どちらもエラトステネスのふるいを用いて素数を求めるC言語のプログラムである。関数 `prime1` は引数で与えられた整数以下のすべての素数を出力する関数であり、ある数が素数かどうかを表現するフラグに `unsigned int` 型を使用し、フラグが立った状態を1、降ろした状態を0で表している。

ここで `unsigned int` 型は4byteとする。ひとつのフラグを表現するには1bitで十分であるため、`unsigned int` 型でフラグを表現するには無駄が多い。そこで、`unsigned int` 型の変数を32個のフラグとして使用することにした。関数 `prime2` はそのように改造したものである。

以下の問いに答えなさい。

- (1) 図1において、関数 `prime1` が引数10で呼び出されたとき、関数 `check_flag` は0以外の値を何回返すか答えなさい。なお、関数 `prime1` は関数 `check_flag` を2箇所呼び出している。
- (2) 図1において、関数 `prime1` が引数100で呼び出されたとき、関数 `reset_flag` が何回呼び出されるかを、求め方とともに答えなさい。
- (3) 図2において、関数 `prime2` が引数1000で呼び出されたとき、関数 `malloc` で割り当てられるメモリーが何byteかを、求め方とともに答えなさい。
- (4) 図2において、関数 `reset_flag` は引数 `p` で表された数に対応するフラグを0にする関数である。(ア)の部分に1文を追加して、関数を完成させなさい。
- (5) 図2において、関数 `prime2` が引数10で呼び出されたとする。関数 `prime2` の終了直前の `flag[0]` の値を16進数で答えなさい。

なお、関数 `ceil` は天井関数で、`double ceil(double x);` のようにプロトタイプ宣言されており、`x` 以上の最小の整数の値を `double` 型で返す。またC言語が提供している `bit` 演算子は以下の通りである。表中の例は4bitでの演算を示している。

演算子	意味	例
<code>&</code>	ビット毎の AND	<code>0011 & 0101 → 0001</code>
<code> </code>	ビット毎の OR	<code>0011 0101 → 0111</code>
<code>^</code>	ビット毎の XOR	<code>0011 ^ 0101 → 0110</code>
<code>~</code>	ビット毎の NOT	<code>~ 0011 → 1100</code>
<code><<</code>	左シフト	<code>0011 << 1 → 0110</code>
<code>>></code>	右シフト	<code>0011 >> 1 → 0001</code>

次ページへ続く

```

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

unsigned int check_flag(unsigned int *flag, unsigned int p) {
    return flag[p];
}

void reset_flag(unsigned int *flag, unsigned int p) {
    flag[p] = 0;
}

void prime1(unsigned int max) {
    unsigned int p, i;
    unsigned int *flag =
        (unsigned int *) malloc(sizeof(unsigned int) * (max + 1));

    // initialization of all flags
    for (p = 2; p <= max; p++) flag[p] = 1;

    // sieving (FURUI NI KAKERU)
    for (p = 2; p * p <= max; p++) {
        if (check_flag(flag, p) != 0) {
            for (i = p * p; i <= max; i += p) reset_flag(flag, i);
        }
    }

    // output
    for (p = 2; p <= max; p++)
        if (check_flag(flag, p) != 0) printf("%d ", p);
    printf("\n");
}

```

図 1: 素数を求める関数 prime1

[次ページへ続く](#)

```

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

unsigned int check_flag(unsigned int *flag, unsigned int p) {
    return (flag[p / 32] & (1 << (p % 32)));
}

void reset_flag(unsigned int *flag, unsigned int p) {
    (ア)
}

void prime2(unsigned int max) {
    unsigned int p, i;
    unsigned int words = ceil((double) (max + 1) / 32);
    unsigned int *flag =
        (unsigned int *) malloc(sizeof(unsigned int) * words);

    // initialization of all flags
    for (i = 0; i < words; i++) flag[i] = 0xFFFFFFFF;

    // sieving (FURUI NI KAKERU)
    for (p = 2; p * p <= max; p++) {
        if (check_flag(flag, p) != 0) {
            for (i = p * p; i <= max; i += p) reset_flag(flag, i);
        }
    }

    // output
    for (p = 2; p <= max; p++)
        if (check_flag(flag, p) != 0) printf("%d ", p);
    printf("\n");
}

```

図 2: 素数を求める関数 prime2

問題4 情報基礎 (2)

図1に示すようなグラフ G_1 、 G_2 、 G_3 をC言語では配列及び構造体を使って表現することが可能である。

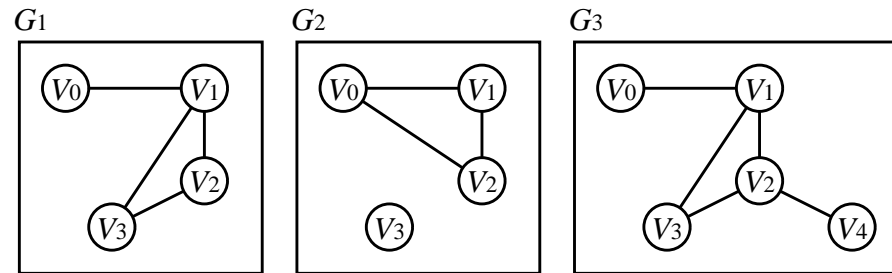


図1

ここでは下記に示す構造体 `node` の配列を用いるものとする。

```
struct node {
    int adjacent;
    struct node *next;
};
```

この構造体 `node` の配列を用いると図1のグラフ G_1 は図2のように表現される。

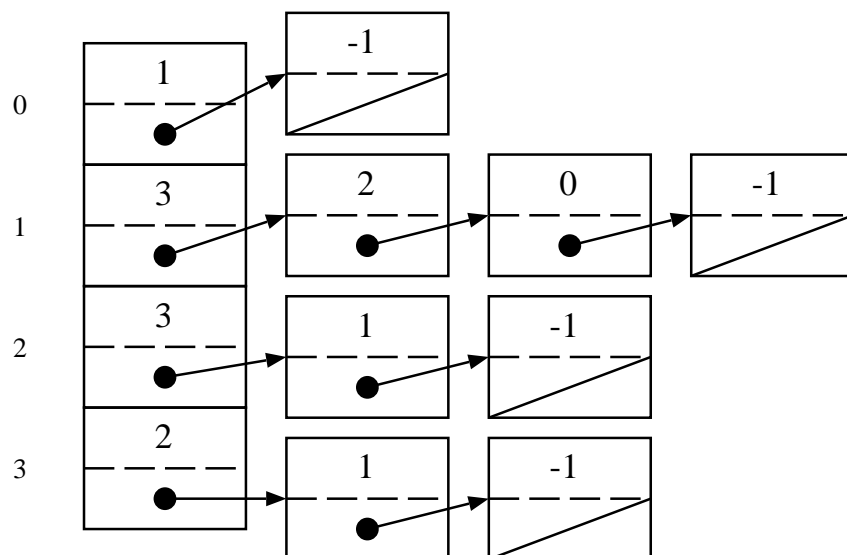


図2

次ページに続く

すなわち、 N 個 ($N \geq 1$) の頂点 V_0, \dots, V_{N-1} からなるグラフ G において、頂点 $V_i (0 \leq i \leq N-1)$ と M 個 ($M \geq 1$) の頂点 $V_{v_0}, \dots, V_{v_{M-1}}$ との間に辺が存在する時、 G を図 3 のように表現する。ただし、図中の -1 は接続している頂点がこれ以上ないことを意味し、斜線は NULL を示す。

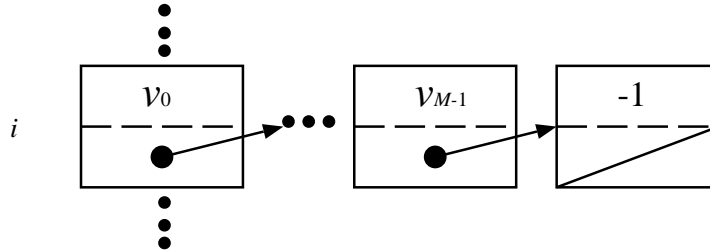


図 3

この構造体 `node` を用いて表現されるグラフ、および C 言語で書かれた以下のコードに関する設問に答えなさい。

C 言語で書かれたコード

```
int is_connected(int nnode, struct node nodes[]) {
    int i, *visited;
    visited = (int *)malloc(sizeof(int)*nnode);
    for (i = 0; i < nnode; i++)
        visited[i] = 0;
    visit(0, visited, nodes);
    for (i = 0; i < nnode; i++)
        if (visited[i] == 0)
            return 0;
    return 1;
}

void visit(int id, int visited[], struct node nodes[]) {
    struct node *n;
    if (visited[id])
        return;
    visited[id] = 1;
```

次ページに続く


```

    for (n = &nodes[id]; n->adjacent != -1; n = n->next)
        visit(n->adjacent, visited, nodes);
}

```

```

int is_euler(int nnode, struct node nodes[]) {
    int i, j, count = 0;
    struct node *n;
    if (!is_connected(nnode, nodes))
        return 0;
    for (i = 0; i < nnode; i++) {
        n = nodes[i].next;
        j = 0;
        while (n != NULL) {
            j++;
            n = n->next;
        }

```

(a)

```

    }
    return count == 0 || count == 2;
}

```

- (1) 関数 `is_connected` にグラフ G を適用する場合、関数の引数としてグラフ G の頂点数とグラフ G を表現した構造体 `node` の配列を与える。この関数 `is_connected` に、図 1 のグラフ G_1 、 G_2 、 G_3 を適用したとき、それぞれのグラフに対する戻り値を答えなさい。
- (2) 関数 `is_connected` に、頂点数 3 のグラフを適用したところ、戻り値として 1 を返した。このようなグラフ 1 つを、図 1 にならって描きなさい。
- (3) 関数 `is_connected` に、辺の数が y であるグラフ G を適用したところ、戻り値として 1 を返した。このとき、関数 `is_connected` の呼び出し開始時から終了時まで、関数 `visit` を呼び出した回数を、 y を用いた式で答えなさい。
- (4) 関数 `is_euler` は、引数としてグラフ G の頂点数と、グラフ G を表現した構造

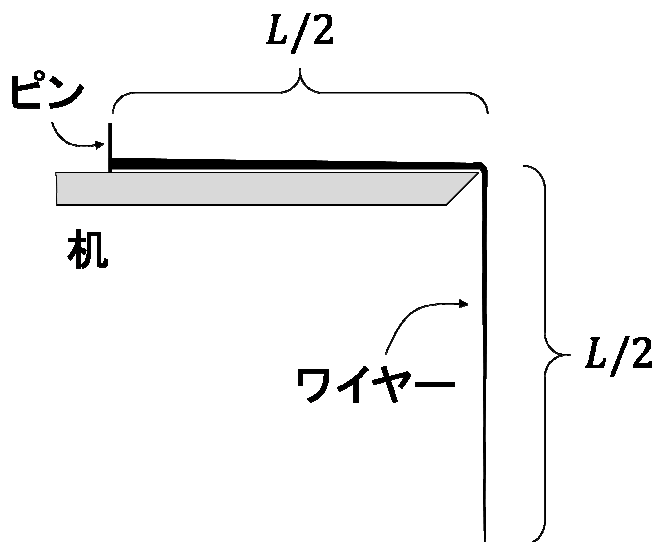
次ページに続く

体 `node` の配列を受け取り、戻り値として、グラフ G が一筆書き可能なら 1 を、そうでなければ 0 を返す。空欄 (a) を埋めて、関数 `is_euler` を完成させなさい。ただし、グラフ G が一筆書き可能である必要十分条件は、グラフ G を関数 `is_connected` に適用するとその戻り値が 1 であること、かつ、グラフ G の頂点のうち、接続している辺の本数が奇数であるものが 0 個または 2 個であることである。

問題 5 物理学 (1)

下図に示すように、長さ L の細くて柔らかいワイヤーが、滑らかで水平な机面上上にのせられている。ワイヤーの左端はピンで固定されており、右側の半分は机の端から垂らされている。ワイヤーの断面は円形で、その半径 r は、下端からのワイヤーの長さ l ($0 \leq l \leq L$) に応じて $r = cl$ となっている。ただし、 c は定数である。ワイヤーの密度を ρ 、重力加速度を g 、机面を高さの基準として、下記の各設問に答えなさい。

- (1) ピンで固定されている状態のワイヤー全体の位置エネルギー U_0 を、 L, c, ρ, g を用いて表しなさい。
- (2) ワイヤー全体の質量を、 c, ρ, L を用いて表しなさい。
- (3) ワイヤーをピンから外すとワイヤーが落下し始める。ピンとワイヤー左端との距離を x ($x \leq L/2$) とするとき、全体の位置エネルギー U_x を L, c, ρ, g, x を用いて表しなさい。
- (4) ピンとワイヤー左端との距離が x のときの運動エネルギー K_x を、 L, c, ρ, \dot{x} を用いて表しなさい。ただし、 \dot{x} は x の時間変化を表す。
- (5) ピンとワイヤー左端との距離が x のときの速度を、 L, g, x を用いて表しなさい。
- (6) $x = L/2$ のときの速度を、 L, g を用いて表しなさい。



問題6 物理学 (2)

以下の問いに答えなさい。

- (1) 図 1(a)のように、真空中にある長方形平行板コンデンサー(奥行き方向の長さ b)がわずかに傾き、板間隔が一方の端で $2(p-\delta)$ 、他方の端で $2(p+\delta)$ になっている ($0 < \delta$, $\delta \ll p$)。このときの静電容量について考える。真空の誘電率を ϵ_0 とし、端の影響は無視する。

- ① 図 1(b)のように座標をとる。問題のコンデンサーを x 軸方向に区切った微小なコンデンサーの静電容量 dC を ϵ_0 、 b 、 dx 、極板間隔 $h(x)$ を使って示しなさい。 dx は十分に小さいものとする。
- ② コンデンサーの全体の静電容量 C を求めなさい。極板の横方向の長さは、 x 軸上に投影したとき $2a$ とする。

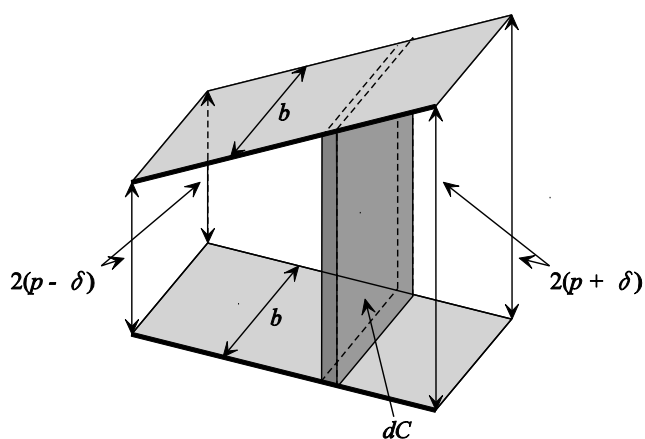


図 1(a)

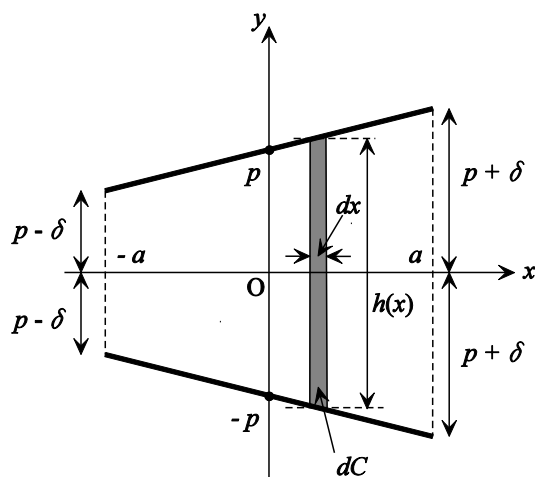


図 1(b) 図 1(a)を手前から見た図

次ページに続く

(2) 図 2 のように、真空中に無限に長い直線導線 ℓ がある。そのわきに一辺の長さが a の正三角形の導線でできた閉回路 ABC が直線導線 ℓ に接している。閉回路の巻き数は 1 回である。直線導線 ℓ と閉回路 ABC は電氣的に接続されていない。また ℓ と辺 BC が平行になるように置かれている。真空の透磁率を μ_0 とする。

- ① 直線導線 ℓ に電流 I が流れているとする。 ℓ から r だけ離れた点 P での磁界の大きさ $H(r)$ を求めなさい。
- ② 閉回路 ABC を貫く磁束 Φ を求めなさい。また、この結果から、相互インダクタンス $M = \Phi/I$ を求めなさい。

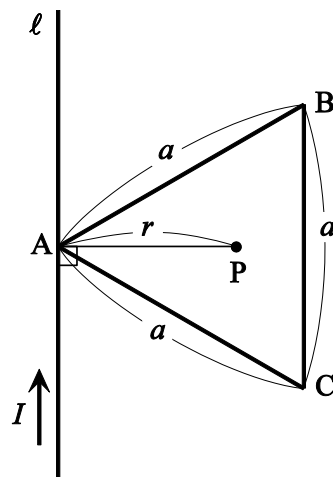


図 2

筑波大学 情報学群 情報科学類

平成 26 年度 学群編入学試験

学力検査問題(外国語)

[注意事項]

1. 試験開始の合図があるまで、問題の中を見てはいけません。
2. 解答用紙の定められた欄に、学群、学類、氏名、受験番号を記入すること。
3. この問題冊子は全部で 2 ページ(表紙、白紙を除く)です。
4. 解答用紙は、合計 1 枚を用いること。

外国語（英語）

次の英文を読んで下の問いに答えなさい。ただし、解答にあたっては、解答用紙に設問番号（(1)、(ア) など）を明記したうえでその設問の解答を記入しなさい。

問題の英文は著作物からの引用であるため、電子的に公開するファイルからは削除しました。

(Life in Simulation, Communications of the ACM, Vol. 56 No. 2, 2013 より一部変更して引用.)

biologist: 生物学者	awash: いっぱいの、あふれて	organism: 有機体
sequence: (ゲノムの) 配列決定	wring: 引き出す	embryo: (生物の) 胚
toxin: 毒素	tissue: 組織	cancerous: 癌にかかった
tractable: 扱いやすい	comprehensive: 網羅的な	bacterium: バクテリア
Mycoplasma genitalium: マイコプラズマ・ゲニタリウム	parasite: 寄生虫	
urethra: 尿道	microbe: 微生物	transcribe: 転写する
differential equation: 微分方程式	Boolean: ブール代数の	glean: 収集する
metabolic: 代謝の		

- (1) ~ に最もよく当てはまる単語を下から選び、A~H の記号で答えなさい。ただし、同一の単語が二つ以上の箇所に当てはまることはないものとする。

A. from	B. to	C. on	D. in
E. for	F. of	G. with	H. without

- (2) 下線部 (イ)、(エ) を和訳しなさい。

- (3) 次の単語を並べ替え、下線部 (ウ) の和文の意味と合うように英文を完成させなさい。

broke build cell down functions individual
into model team the the their to various

- (4) 下線部 (ア) の原因と、そのデータを使って科学者がどのようなことをしようとしているのかを、簡潔に 100 字程度で述べなさい。

- (5) 下線部 (オ) について、シミュレーションを実行した結果がどのように役立つかを簡潔に 150 字程度で述べなさい。