

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 30 年度 学群編入学試験

学力試験問題(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. 解答用紙と下書き用紙の定められた欄に、氏名、受験番号を記入すること。
3. この問題冊子は全部で 10 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) xyz 直交座標系において円柱面 $x^2 + y^2 = 1$, xy 平面, 平面 $z = x$ により囲まれた部分の体積を求めなさい.
- (2) 関数 $f(x, y)$ は, x および y について偏微分可能で $x \frac{\partial f}{\partial x} + y \frac{\partial f}{\partial y} = 0$ なる関係を満足する. 関数 $f(x, y)$ を $x = r \cos \theta$, $y = r \sin \theta$ ($r > 0, 0 \leq \theta < 2\pi$) で変数変換したときの $f(r \cos \theta, r \sin \theta)$ は, 変数 r を含まない関数となることを証明しなさい.

問題2 数学(2)

二つのメーカー X および Y からなる市場において、各メーカーのユーザー数を調査したい。毎年メーカー X のユーザーのうち $\frac{1}{10}$ がメーカー Y のユーザーとなり、一方で、メーカー Y のユーザーのうち $\frac{1}{5}$ がメーカー X のユーザーとなる。それ以外は同じメーカーのユーザーのままでいるものとし、ユーザーの総数は変化しない。このとき以下の問いに答えなさい。

- (1) ある年におけるメーカー X, Y のユーザー数をそれぞれ x_n, y_n で表す。このとき翌年におけるそれぞれのメーカーのユーザー数 x_{n+1}, y_{n+1} を二次正方行列 A を使って以下の形で表す。行列 A を具体的に示しなさい。

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = A \begin{pmatrix} x_n \\ y_n \end{pmatrix}$$

- (2) A の固有値および固有ベクトルを求めなさい。
- (3) $P^{-1}AP$ が対角行列となるような行列 P を一つ求めるとともに、 P の逆行列 P^{-1} を求めなさい。
- (4) 行列 A^n を求めなさい。
- (5) (4) の結果を使って、 $n \rightarrow \infty$ としたときのメーカー X および Y のユーザー数の比率を求めなさい。

問題3 情報基礎 (1)

図1の例のように、各行に学籍番号、試験を受けた科目名、試験の得点が記録されたファイルdata.txtがあるとする。例えば、この例の1行目では、10001という学籍番号の学生がMathという科目の試験を受け、95点の得点をとったことを表している。

ファイルに記録される科目名は19文字以下の長さの英文字の文字列であり、その文字列には空白文字は含まれないものとする。また、得点は0点から100点までの整数であるとする。学籍番号と科目名、科目名と得点の間にはそれぞれ空白文字が1文字入っている。

```
10001 Math 95
10003 Science 90
10002 Science 83
10005 Math 68
10002 Math 86
10004 Science 75
10001 Science 100
10002 English 92
.
.
.
```

図1 data.txtの例

このファイルに格納されたデータに対していくつかの処理を行うC言語の関数について考える。これらの関数において、ファイルのデータを格納するために以下の構造体dtと大域変数headを用いる。

```
struct dt {
    int student;
    char course[20];
    int score;
    struct dt *next;
};

struct dt *head;
```

構造体dtのstudentには学籍番号、courseには科目名、scoreには得点をそれぞれ格納する。この構造体を用いて図2に示すような線形リストを構成し、nextにはリストの次の要素へのポインタを格納する。図2で「NULL」はNULLポインタを表している。

変数headにはこの線形リストの最初の要素へのポインタを格納する。

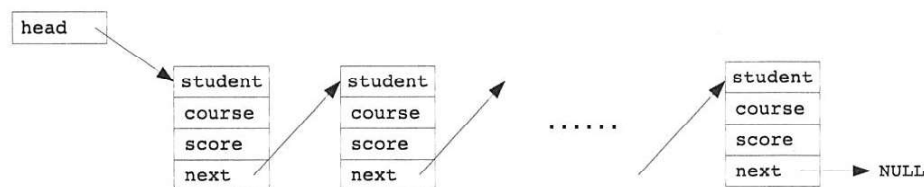


図2 線形リスト

次のページに続く

以下の(1)~(5)に答えなさい。

- (1) 関数 load_file はファイル data.txt の内容を読み込んで図2のような線形リストを構成するものである。

空欄 (a), (b) を埋めてこの関数を完成させなさい。ただし、空欄 (a) には C 言語の式が一つ入り、空欄 (b) には C 言語の代入文が一つ入る。

なお、関数 fscanf はファイルから指定された書式にしたがってデータを読み込んで変数に格納するものである。関数 load_file ではファイル data.txt をオープンして fscanf でデータを読み込んでいるが、この fscanf の最初の呼び出しではファイルの1行目の「10001」「Math」「95」が変数 s、配列 c、変数 sc にそれぞれ格納される。ファイルの最後まで読み込みが終わると fscanf は EOF を返す。

```
void load_file()
{
    FILE *file = fopen("data.txt", "r");
    int s;
    char c[20];
    int sc;

    head = NULL;

    while (fscanf(file, "%d %s %d", &s, c, &sc) != EOF) {
        struct dt *new;
        new = (struct dt *) malloc(sizeof(struct dt));
        new->student = s;
        strcpy(new->course, c);
        new->score = sc;

        new->next = 

|     |
|-----|
| (a) |
|-----|

;
        

|     |
|-----|
| (b) |
|-----|

;
    }
}
```

- (2) 関数 load_file の説明として正しいものを以下からひとつ選びなさい。

- (A) head の指す線形リストには、data.txt に記録されているのと同じ順番に要素が並べられる。
- (B) head の指す線形リストには、data.txt に記録されているのと逆の順番に要素が並べられる。
- (C) head の指す線形リストには、data.txt に記録されているのとは同じ順番でも逆の順番でもない順番に要素が並べられる。

次のページに続く

- (3) 関数 `remove_one` は、引数に与えられたポインタ `d` が指す要素を `head` が指す線形リストから取り除く (ただし、線形リストから取り除いた要素のメモリは解放しない) ものである。なお、`head` は `NULL` ではないものとする。

空欄 (c), (d) を埋めてこの関数を完成させなさい。ただし、空欄 (c) には C 言語の代入文が一つ入り、空欄 (d) には C 言語の式が一つ入る。

```
void remove_one(struct dt *d)
{
    struct dt *p, *prev;

    if (head == d) {
        (c);
        return;
    }

    prev = head;
    for (p = head->next; p != NULL; prev = p, p = p->next) {
        if (p == d) {
            prev->next = (d);
            return;
        }
    }
}
```

- (4) 関数 `sort_list` は、`head` が指す線形リストの要素を入れ替えて、リストの要素が学籍番号の小さい順 (昇順) になるように並べ替えるものである。

空欄 (e)~(g) を埋めてこの関数を完成させなさい。ただし、空欄 (e) には C 言語の式が一つ入り、空欄 (f), (g) には C 言語の代入文がそれぞれ一つ入る。

```
void sort_list()
{
    struct dt *p;
    struct dt *min = head;

    if (head == NULL) {
        return;
    }

    for (p = head->next; p != NULL; p = p->next) {
        if (min->student > p->student) {
            min = (e);
        }
    }

    remove_one(min);
    sort_list();
    (f);
    (g);
}
```

次のページに続く

- (5) 関数 `print_average` は、関数 `sort_list` によって並べ替えられた `head` が指す線形リストのそれぞれの学生について、学籍番号、試験を受けた科目数、試験を受けたすべての科目の得点の平均点 (小数点以下切り捨て) を図 3 のような形式で関数 `printf` を用いて出力するものである。

空欄 (h) を埋めてこの関数を完成させなさい。ただし、変数 `count` は個々の学生が試験を受けた科目数を数えるために、変数 `total` は試験を受けた科目の得点の合計を求めるために、それぞれ用いること。また、空欄 (h) の中で新たな変数を宣言しないこと。

```
void print_average()
{
    int count = 0, total = 0;
    struct dt *p;

    for (p = head; p != NULL; p = p->next), {
        (h)
    }
}
```

```
10001 2 97
10002 3 87
.
.
.
```

図 3 `print_average` の出力例

問題 4 情報基礎 (2)

昇順に整列された C 言語の `int` 型配列の中から与えられた数値 (以下、探索キーと呼ぶ) と一致する配列要素を二分探索を用いて見つけ、その配列要素の添字を求めることを考える。以下の小問に答えなさい。

- (1) C 言語で書かれた `binsearch1` という以下の関数は、`int` 型の配列 `a[]`、配列 `a[]` の大きさ `n`、探索キー `key` を与えられて、`a[0]` から `a[n-1]` までの配列の中で探索キーと一致する配列要素の添字を返す関数である。ただし、`a[]` は昇順に整列されているものとする。配列中に探索キーと同じ数値が複数存在した場合、それらのうちいずれか一つの添字を返す。また、探索キーと一致する数値が存在しない場合は `-1` を返す。以下の (ア) および (イ) の空欄に入れるべき条件式を答え、関数 `binsearch1` を完成させなさい。

```
int binsearch1(int a[], int n, int key){
    int left = 0, right = n - 1, mid;
    while(left <= right){
        mid = left + (right - left) / 2;
        if( (ア) ) right = mid - 1;
        else if( (イ) ) left = mid + 1;
        else return mid;
    }
    return -1;
}
```

- (2) 関数 `binsearch1` のような二分探索と線形探索 (配列を先頭から順に探索キーと比較し、一致する値を探す方法) の時間計算量を漸近的計算量 (オーダー) などを用いて比較しなさい。また、`binsearch1` がそのような計算量になる理由を述べなさい。
- (3) 探索キーと一致する数値が配列中に複数存在する場合に、該当する添字のうち最も小さな添字を求めたい。以下の関数 `binsearch2` は `binsearch1` を利用して最も小さな添字を見つけて返す関数である。配列 `a[]` の大きさ `n` が十分に大きいとき、この関数の時間計算量が最悪となるデータの例を挙げなさい。また、そのデータ例で何故最悪になるかの理由を述べなさい。

次ページへ続く


```

int binsearch2(int a[], int n, int key){
    int i = binsearch1(a, n, key), j;
    if(i != -1){
        for(j = i; j >= 0; j--){
            if(a[j] < key) return j + 1;
        }
        return 0;
    } else return -1;
}

```

- (4) binsearch1 は検索キーと一致する数値を見つけると直ちに終了しその添字を返した。しかし、一致する数値を見つけても終了せずに二分探索を続けることにより、探索キーと一致する複数の配列要素の添字のうち最も小さい添字を返しかつ binsearch2 よりも効率のよい関数 binsearch3 が書ける。以下に示すプログラムの (ウ) および (エ) の空欄を埋めて binsearch2 と同じ働きをしかつ binsearch2 よりも効率のよい関数 binsearch3 を完成させなさい。ただし、(ウ) には条件式、(エ) には C 言語の式が 1 つ入る。また、(ア) および (イ) の空欄には小問 (1) と同じものが入る (小問 (4) では答えなくてよい)。

```

int binsearch3(int a[], int n, int key){
    int left = 0, right = n - 1, mid;
    while(left <= right){
        mid = left + (right - left) / 2;
        if( (ア) ) right = mid - 1;
        else if( (イ) ) left = mid + 1;
        else if( (ウ) ) return mid;
        else (エ) ;
    }
    return -1;
}

```

- (5) 関数 binsearch3 と binsearch2 の時間計算量を漸近的計算量 (オーダー) などを用いて比較しなさい。また、binsearch3 がそのような計算量になる理由を述べなさい。

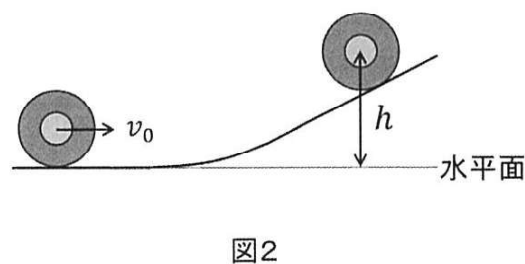
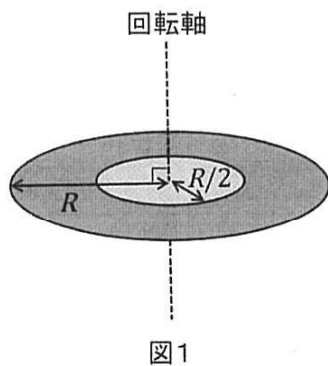
問題5 物理学(1)

図1のように質量 m 、半径 R の薄い円盤を考える。この円盤の面密度(単位面積あたりの質量)は半径 $R/2$ を境に異なっており、内側の領域の面密度は外側の領域の面密度の α 倍とする。ただし、 $\alpha \geq 0$ とし、それぞれの領域の面密度は一定とする。このとき、以下の問いに答えなさい。

- (1) 内側の領域と外側の領域の面密度を求めなさい。
- (2) 円盤の中心から垂直に回転軸をとる。このとき回転軸まわりの円盤の慣性モーメントを求めなさい。

次に、この円盤を図2のように鉛直に立てて質量中心の初速度 v_0 で水平面上を転がし、その後、斜面を上がっていく状況を考える。この際、円盤はすべらずなめらかに転がるものとする。また、重力加速度は g とし、円盤に空気抵抗は働かないものとする。このとき、以下の問いに答えなさい。

- (3) 円盤の速度が0になるときの円盤の質量中心の高さ h を求めなさい。ただし、水平面の高さを0とする。
- (4) h が最大になる α と、その時の h を求めなさい。



問題 6 物理学 (2)

以下の問いに答えなさい。

- (1) 真空中で、電荷量 q_1 , q_2 , q_3 の 3 つの点電荷が、正三角形の各頂点の位置にある。なお、正三角形の 1 辺の長さは r とし、真空の誘電率は ϵ_0 とする。
- ① 正三角形の重心の電位を求めなさい。なお、無限遠の電位を 0 とする。
 - ② 3 つの点電荷による系の静電 (電気) ポテンシャルエネルギーを求めなさい。なお、3 つの点電荷が互いに無限遠にある状態の静電 (電気) ポテンシャルエネルギーを 0 とする。

- (2) 図 1 に示すように、真空中で、磁束密度 \vec{B} の一定で一様な磁場 (紙面に垂直) の中に置かれた導線に一定値 I の電流が流れている (導線は紙面内に固定されている)。導線は、原点 O を中心とした半径 a の半円導線と、その両端に接続された x 軸上の長さ l の 2 つの直線導線からなる。なお、真空の透磁率は μ_0 とする。

- ① 一定値 I の電流により、原点 O に生成される磁束密度の大きさを求めなさい。
- ② 磁束密度 \vec{B} の磁場により、導線全体が受ける合力の大きさを求めなさい。

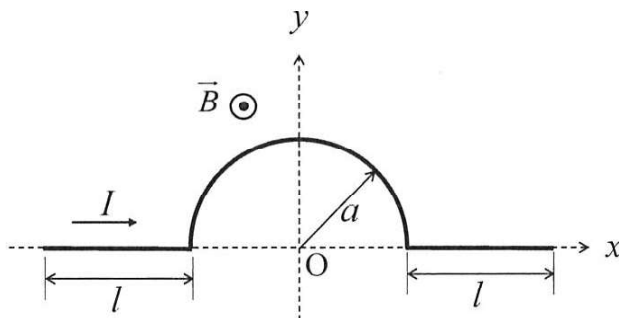


図 1