

# 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 29 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 11 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) 関数  $f(x) = e^x$  をマクローリン展開 ( $x = 0$  のまわりでテイラー展開) せよ。

(2) 以下の性質 (A) を用いて、次の極限值を求めよ。

$$\lim_{n \rightarrow \infty} \frac{1}{n} \left( \frac{n}{1!} + \frac{n-1}{2!} + \cdots + \frac{2}{(n-1)!} + \frac{1}{n!} \right)$$

(A) 数列  $\{a_n\}_{n=1}^{\infty}$  に対して、 $\lim_{n \rightarrow \infty} a_n = \alpha$  のとき、

$$\lim_{n \rightarrow \infty} \frac{a_1 + \cdots + a_n}{n} = \alpha$$

が成り立つ。

(3) 上の性質 (A) を証明せよ。

## 問題 2 数学 (2)

実ベクトル空間  $W$  のベクトル  $\mathbf{v}_1, \dots, \mathbf{v}_n$  が次の 2 つの条件を満たしているものとする:

(A)  $\|\mathbf{v}_1\| = \dots = \|\mathbf{v}_n\| = 1$

(B) 相異なる  $j, k \in \{1, \dots, n\}$  に対して  $\langle \mathbf{v}_j, \mathbf{v}_k \rangle = 0$

ただし,  $\langle, \rangle$  は  $W$  の内積,  $\| \cdot \|$  はこの内積で定まる長さを表す. また,  $\mathbf{v}_1, \dots, \mathbf{v}_n$  の 1 次結合によって表されるベクトル全体からなる集合を  $V$  とする. 以下の (i)–(iv) を証明しなさい.

(i)  $\mathbf{v}_1, \dots, \mathbf{v}_n$  は 1 次独立である.

(ii) 任意のベクトル  $\mathbf{x} \in V$  が実数  $x_1, \dots, x_n$  を用いて

$$\mathbf{x} = \sum_{j=1}^n x_j \mathbf{v}_j$$

と表されるとき, 次の等式が成り立つ:

$$\|\mathbf{x}\|^2 = \sum_{j=1}^n x_j^2$$

(iii)  $V$  は  $W$  の部分空間である.

(iv)  $V \neq W$  であれば,  $\mathbf{w} \notin V$  かつ  $\mathbf{w} \in W$  を満たす任意のベクトル  $\mathbf{w}$  に対して  $\mathbf{v}_1, \dots, \mathbf{v}_n, \mathbf{w}$  は 1 次独立である.

### 問題3 情報基礎 (1)

ある学園の試験（数学、英語、各教科 100 点満点）の結果を成績順に出力するプログラムを作成している。ファイルからのデータの読み込み部分、およびメイン関数はすでに作成が終わっている。そのプログラムの一部を以下に示す。各行の先頭は行番号を表している。

```
1 struct item {
2     char id[8]; /* 学籍番号 */
3     int math; /* 数学の点数 */
4     int eng; /* 英語の点数 */
5 };
6
7 #define N 1024 /* このプログラムで取り扱える学生の最大数 */
8 int main(int argc, char *argv[])
9 {
10     struct item table[N]; /* 成績が格納される配列 */
11     int rank[N]; /* 順位に従ってtableの添字が格納される配列 */
12     int size = readfile(N, table);
13     /* ファイルからデータを読み込む。返回值:読み込んだデータ数 */
14     /* ファイルに格納されているデータ数はN以下であるとする。 */
15     make_ranking(size, rank, table);
16     print_table(size, rank, table);
17     return 0;
18 }
```

アキバさんとイバラさんは、プログラムの 15 行目で利用される、成績が格納されている配列（第 3 引数）の添字（インデックス、index）を、数学の点数の低い順に第 2 引数で指定した配列に入れる関数 `make_ranking` を作成した。以下の小問に答えなさい。

次ページに続く

- (1) プログラムの 16 行目で利用される関数 `print_table` は、一行ごとに、数学の点数の低い順の順位、学籍番号、数学の点数、英語の点数を出力する関数である。以下の空欄を埋めて、関数 `print_table` を完成させなさい。

```
1  #include <stdio.h>
2  void print_table(int size, int r[], struct item t[])
3  {
4      int i;
5      for (i = 0; i < size; i++) {
6          printf("%4d %7s %3d %3d\n", i + 1,
7              (a), (b), (c));
8      }
9  }
```

- (2) アキバさんのプログラムを以下に示す。このプログラムが正しく動作するように 14 行目で利用される関数 `swap` を書きなさい。

```
1  int cmp(int i, int j, int r[], struct item t[])
2  {
3      return t[r[i]].math - t[r[j]].math;
4  }
5
6  void make_ranking(int size, int r[], struct item t[])
7  {
8      int i, j;
9      for (i = 0; i < size; i++)
10         r[i] = i;
11     for (i = 0; i < size; i++) {
12         for (j = size - 1; j > i; j--) {
13             if (cmp(i, j, r, t) > 0)
14                 swap(r, i, j);
15         }
16     }
17 }
```

- (3) 小問(2)で示したアキバさんのプログラムを、数学の点数が同じ場合、英語の点数が低い順となるようにするには、1 行目から 4 行目で定義されている関数 `cmp` を書き換えればよい。この要件を満たすように関数 `cmp` を書き直しなさい。

[次ページに続く](#)

- (4) イバラさんのプログラムを以下に示す。イバラさんはこの関数について「(d) 行目から (e) 行目の部分で数学の点数の累積度数を調べ、(e) 行目の次の行からその度数を見て順位を調べる」と説明している。空欄に入る数値をプログラムの行番号を用いて答えなさい。

```
1  #define MAX_SCORE 100
2  void make_ranking(int size, int r[], struct item t[])
3  {
4      int count[MAX_SCORE + 1];
5      int i;
6      for (i = 0; i <= MAX_SCORE; i++)
7          count[i] = 0;
8      for (i = 0; i < size; i++)
9          count[t[i].math]++;
10     for (i = 1; i <= MAX_SCORE; i++)
11         count[i] += count[i - 1];
12     for (i = size - 1; i >= 0; i--)
13         r[--count[t[i].math]] = i;
14 }
```

- (5) イバラさんのプログラムでは、数学の点数が同じ場合、順番は元のデータの順番が維持されるようになっている。小問(3)と同様に数学の点数が同じ場合、英語の点数が低い順となるようにするには、どのような方針で改変すればよいか、2行から5行程度で説明しなさい。具体的なプログラムを示す必要はない。
- (6) 小問(2)で示したアキバさんのプログラムと小問(4)で示したイバラさんのプログラムのどちらを使うかを、計算に要する時間や使用するメモリ量を考慮して考える。アキバさんのプログラムと比較したときの、イバラさんのプログラムの長所と短所を、合わせて4行から10行程度で述べなさい。

#### 問題4 情報基礎 (2)

SNS (Social Networking Service、ソーシャル・ネットワーキング・サービス) における友人関係を無向グラフにより表現する。この無向グラフでは、頂点 (vertex) は、会員と対応している。会員 X と会員 Y が直接友人であることを、会員 X に対応した頂点から会員 Y に対応した頂点への辺 (edge) により表現している。以下にそのような無向グラフの例を示す。

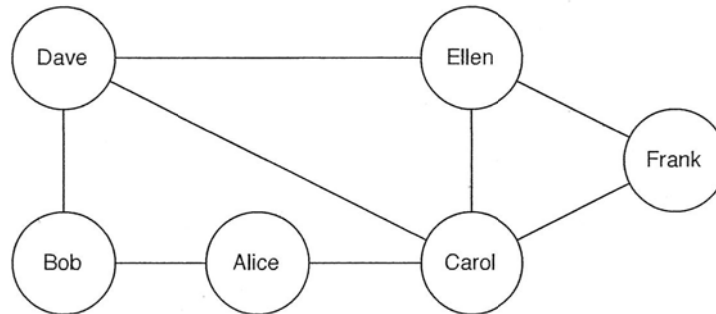


図 1 無向グラフによる友人関係の表現の例

この無向グラフを次の構造体で表現する。

```
struct person {
    char *name;
    struct plist *friends;
};

struct plist {
    struct person *data;
    struct plist *next;
};
```

この無向グラフで頂点は、構造体 person で表現されている。これは、name、および、friends というフィールドを持っている。name は、会員の名前を文字列で保持している。friends は、隣接している頂点をリスト構造で保持している。

(1) 次の関数は、頂点に対応した会員の名前、および、友人の名前を画面に出力する関数である。

```
void print_person( struct person *p ) {
    printf("Name: %s, friends: { ", p->name );
    for( struct plist *pl=p->friends; pl!=NULL; pl=pl->next ) {
        printf("%s, ", pl->data->name );
    }
    printf("}\n");
}
```

次ページに続く

図 1 で頂点 Alice を表す変数を引数としてこの関数を呼び出した。すると、画面に次のような結果が出力された。

Name: Alice, friends: { Carol, Bob, }

図 1 で頂点 Bob を表す変数を引数としてこの関数を呼び出すと画面にどのような結果が出力されるか答えなさい。なお、出力される友人の順序は問わない。

(2) 次の関数は、2名の会員が直接友人であることを示すために、無向グラフに辺を追加する関数である。空欄を埋めて、関数を完成させなさい。

```
void become_friends( struct person *p1, struct person *p2 ) {  
    plist_add( (a) );  
    plist_add( (b) );  
}
```

ただし、この関数は、struct plist \* のデータを操作するために、次の関数を利用している。また、リストへの要素の追加は必ず成功するものとする。

```
void plist_add( struct plist **plp, struct person *p )  
    *plp が指しているリストの先頭に要素 p を追加する。
```

(3) 最初に示したように、この無向グラフを表現するために、各頂点は隣接している頂点をリスト構造で保持している。無向グラフが多数の頂点と辺を持つ時、無向グラフを表現するために必要なメモリ量（空間的な複雑さ）のオーダーを答えなさい。ただし、無向グラフに含まれる頂点の数を  $n$ 、辺の数を  $m$  とし、これらを必要に応じて使いなさい。

(4) 次の関数は、2名の会員の共通の友人を画面に表示する関数である。空欄を埋めて完成させなさい。ただし、利用しない空欄には「/\*\*/」と書きなさい。

```
void print_common_friends( struct person *p1, struct person *p2 )  
{  
    (c)  
    printf("%s and %s have common friends: { ",  
           p1->name, p2->name );  
    for( (d) ) {  
        for( (e) ) {  
            if( (f) )  
                printf("%s, ", (g) );  
        }  
    }  
    printf("}\n");  
}
```

次ページに続く



(5) 小問(4)の関数は、2重ループを含むので友人の数が増えると急激に効率が悪くなるという問題がある。この問題を解決する方法の概略を2～5行程度で述べなさい。なお、具体的なプログラムを示す必要はない。また、データ構造を変更しても良い。

(6) SNS では、ある会員が自分の情報を限定した他の会員にのみアクセスさせたいことがある。たとえば、直接の友人、友人の友人、友人の友人の友人などに限定してアクセスさせたいことがある。無向グラフのある頂点と頂点を結ぶ辺の数の最小を距離と定義する。ある会員の情報を距離がある数以下の友人に限りアクセスさせたいとする。

次の関数 `access_ok()` は、距離 `maxhop` 以下でつながりがある会員を待ち行列を使って幅優先 (`breadth-first`, 横型) で探索し、その範囲内で見つかれば1、見つからなければ0を返す関数である。空欄を埋めて、関数を完成させなさい。ただし、利用しない空欄には「`/**/`」と書きなさい。これらの関数で待ち行列の割り当て、および、リストと待ち行列への要素の追加は必ず成功するものとする。

```
int access_ok( struct person *p1, struct person *p2, int maxhop )
{
    struct plist *visited = NULL;      /* 空のリストを作成する。*/
    struct pqueue *q = pqueue_alloc(); /* 待ち行列を割り当てる。*/
    int result = access_ok1( p1, p2, &visited, q, maxhop );
    pqueue_free( q );                  /* 待ち行列 q を解放する。*/
    plist_free( visited );              /* リストを解放する。*/
    return( result );
}

void plist_free( struct plist *pl ) {
    リストを解放する。内容省略。
}

void plist_add( struct plist **plp, struct person *p ) {
    *plp が指しているリストの先頭に要素 p を追加する。内容省略。
    小問(2) のものと同じ。
}

int plist_contains( struct plist *pl, struct person *p ) {
    リスト pl に要素 p が含まれていたら 1、含まれていなければ 0 を返す。
    内容省略。
}
```

次ページに続く

```

struct pqueue *pqueue_alloc() {
    struct person * 型の値、および、整数を保持する待ち行列を割り当て、
    そのポインタを返す。内容省略。
}

void pqueue_free(struct pqueue *q) {
    待ち行列を解放する。内容省略。
}

void pqueue_put( struct pqueue *q, struct person *p, int n ) {
    待ち行列 q に要素 p 、および、 n を追加する。内容省略。
}

int pqueue_get( struct pqueue *q, struct person **pp, int *np )
{
    待ち行列 q が空の時は、何もしないで 0 を返す。
    待ち行列 q が空ではない時は、struct person * 型と整数型の要素を取り
    出し、それぞれを *pp と *np へセットする。結果として 1 を返す。
    内容省略。
}

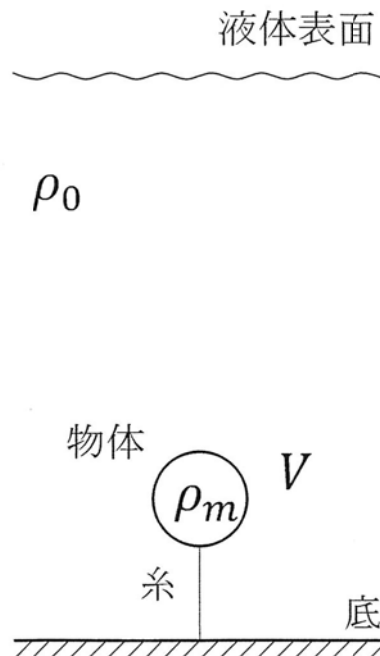
int access_ok1( struct person *p1, struct person *p2,
    struct plist **visited, struct pqueue *q, int maxhop ) {
    struct person *p; int hop;
    pqueue_put( q, p1, 0 ); /* 待ち行列に p1 と 0 を追加する。*/
    while( pqueue_get(q, &p, &hop) ) { /* 待ち行列から struct
                                           person *と整数を取り出す。*/
        (h) ;
        if( (i) )
            return( 0 );
        if( (j) )
            return( 1 );
        for( struct plist *pl=p->friends; pl!=NULL; pl=pl->next )
        {
            struct person *f = pl->data;
            if( (k) )
                (l) ;
        }
    }
    return( 0 );
}

```

### 問題5 物理学(1)

下図に示すように、密度  $\rho_0$  の静止した液体中の十分深い底の位置に、体積  $V$ 、密度  $\rho_m$  の物体が十分軽い糸で固定されている。ただし、 $\rho_0 > \rho_m$  である。重力加速度の大きさを  $g$  として以下の問いに答えなさい。

- (1) 糸に働いている張力の大きさを  $\rho_0$ ,  $V$ ,  $\rho_m$ ,  $g$  を用いて表しなさい。
- (2) 糸を切った瞬間の物体の加速度の大きさを  $\rho_0$ ,  $\rho_m$ ,  $g$  を用いて表しなさい。
- (3) 糸を切ると物体は浮上し始める。物体は速さに比例する抵抗力を受け、その比例定数を  $k$  とする。物体の速さは  $u_{\max} = \frac{\rho_m V g}{k} \left( \frac{\rho_0}{\rho_m} - 1 \right)$  を超えないことを示しなさい。ただし、物体は鉛直上向きに浮上し、液体表面に達しないものとする。
- (4) 糸を切ってから物体の速さが  $u_{\max}$  の  $\alpha$  倍になるまでに浮上した距離を  $\rho_0$ ,  $V$ ,  $\rho_m$ ,  $g$ ,  $k$ ,  $\alpha$  を用いて表しなさい。ただし、物体は鉛直上向きに浮上し、液体表面に達しないものとする。また、 $\alpha < 1$  とする。



## 問題6 物理学(2)

図1(a)のように、極板間隔が $a$ 、面積が $S$ の平行平板コンデンサーがある。極板に垂直な座標軸 $x$ をとる。その極板間に、 $x=0$ で誘電率が $\epsilon_1$ となり、 $x=a$ で $\epsilon_2$ となる誘電体を満たす。誘電率は $x$ に対して線形に増加する。次の順序に従って、図1(a)のコンデンサーの静電容量を求めることにする。ただし、各極板の端の影響は無視する。

(1) 次の⑦～⑨に入る適切な式を答えなさい。

- ・  $x$  ( $0 \leq x \leq a$ ) における誘電率 $\epsilon(x)$ を $\epsilon_1$ ,  $\epsilon_2$ ,  $a$ ,  $x$ を使って表すと、 $\epsilon(x) = \text{⑦}$ となる。
- ・ 図1(b)のように、極板間隔 $\Delta x$ 、面積 $S$ 、誘電率 $\epsilon(x)$ の平行平板コンデンサーの静電容量を $C_A$ とすると、 $1/C_A$ を $\Delta x$ ,  $S$ ,  $\epsilon(x)$ を使って表すと、 $1/C_A = \text{⑧}$ となる。ただし、 $\Delta x$ は十分小さく、その間の誘電率 $\epsilon(x)$ は一定と見なしてよい。
- ・ 図1(c)のように、静電容量 $C_1$ ,  $C_2$ ,  $C_3$ のコンデンサーを直列接続した。その合成容量を $C_B$ とすると、 $1/C_B$ を $1/C_1$ ,  $1/C_2$ ,  $1/C_3$ を使って表すと、 $1/C_B = \text{⑨}$ となる。

(2) (1)の結果を使って、図1(a)のコンデンサーの静電容量 $C$ を求めなさい。

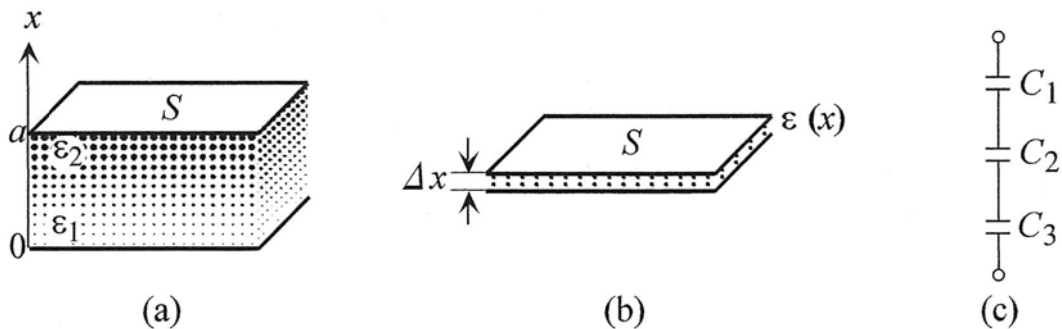


図1