筑波大学　情報学群　情報科学類・情報メディア創成学類

令和４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９ページ（表紙、白紙を除く）です。
4．　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、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。
7．解答用紙の裏面を使用する場合には、その旨を解答用紙の表面に示してください。

問題1　数学1

(1) 次の2重積分について，以下の問いに答えなさい．

$$I = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{f(x,y)} dxdy$$

(1-1) $f(x,y) = -x^2 - y^2$ とし，I を求めなさい．

(1-2) $f(x,y) = -ax^2 - 2bxy - cy^2$ とし，I を求めなさい．ただし $a > 0$，$b^2 - ac < 0$ とする．

(2) 次の方程式が開区間 $(-\sqrt{2}, \sqrt{2})$ で互いに異なる解をちょうど n 個持つことを証明しなさい．

$$\frac{d^n}{dx^n}(x^2 - 2)^n = 0$$

問題２　数学２

(1) 以下に示す実ベクトル空間 $\mathbb{R}^4$ のベクトル $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5$ について，線形独立となる組合せを一つ挙げなさい．このとき，残り全てのベクトルを線形独立なベクトルの一次結合として表しなさい．

$$\boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}, \boldsymbol{v}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ -1 \end{pmatrix}, \boldsymbol{v}_3 = \begin{pmatrix} 2 \\ 1 \\ 3 \\ 5 \end{pmatrix}, \boldsymbol{v}_4 = \begin{pmatrix} 1 \\ -1 \\ 2 \\ 1 \end{pmatrix}, \boldsymbol{v}_5 = \begin{pmatrix} 3 \\ 1 \\ 6 \\ 5 \end{pmatrix}$$

(2) 実ベクトル空間 $\mathbb{R}^3$ の部分集合 W について答えなさい．

$$W = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \, x^2 - y^2 + 4z^2 + 4xz = 0 \right\}$$

(2–1) W は $\mathbb{R}^3$ の二つの部分空間の組合せで表すことができる．それぞれを W_1, W_2 とするとき，W_1, W_2 を求めるとともに，W を W_1, W_2 を使って表しなさい．

(2–2) W は部分空間かどうかを理由とともに答えなさい．

問題３　情報基礎１

取りうる値の範囲があらかじめ定まっている整数のデータを整列させるアルゴリズムに関する以下の問いに答えなさい．

(1)ビンソート ビンソートは，まずデータが取りうる値すべてに対応する箱（ビン）を１つずつ用意し中身を空にする．そして，データ１つ１つをその値に対応する箱に代入する．すべてのデータを箱に入れ終えたあと，小さい値に対応する箱から順番に中身を取り出せば，昇順に並べられたデータを得ることができる．

以下の関数 bin_sort は，上述のビンソートのアルゴリズムを C 言語で書いたものである．関数 bin_sort は，データの入った配列 a と配列 a の要素数 n を受け取り，a の要素をビンソートによって昇順に並べ替える．このプログラムに関して以下の問いに答えなさい．配列 a の要素は 0 以上 10 以下の整数とする．

(1-1) この関数では常にデータを正しく整列できるとは限らない．a={3,1,4,8,4}, n=5 をこの関数に与えて実行した後の a の値を書き出しなさい．

(1-2) どのような場合に，この関数でデータを整列させることができないか答えなさい．また，上述のビンソートにおいて正しくデータを整列させるには，この関数をどのように修正すれば良いか説明しなさい．

```
#define M 10
void bin_sort(int a[], int n)
{
  int i, j, bin[M+1];
  for (i=0; i <= M; i++) /* 箱を空にする */
    bin[i] = -1;
  for (i=0; i< n; i++) /* 対応する箱に代入する*/
    bin[a[i]] = a[i];
  j=0;
  for (i=0; i<= M; i++) /* データを箱から昇順に取り出し，配列 a に戻す */
    if (bin[i] != -1)
      a[j++] = bin[i];
}
```

次ページに続く

(2)分布数え上げソート 分布数え上げソートは，1つのデータに1つのキーが付随していると仮定し，このようなデータを整列させる．ここでは，データの値そのものをキーとして採用する．まず，すべてのデータを調べ，キーの出現頻度を求める．キーの出現頻度がわかれば，キーの値から，データを昇順に並べたときにキーがデータの何番目に当たるかを知ることができる．そして，これらの情報をもとに昇順に並べられたデータを取り出す．

例えば，キーの範囲を0以上5以下として，分布数え上げソートでデータの並び 4,3,1,3,5 を整列させる場合，各キーが出現する回数と累計は表1の分布表のようになる．キーkを持つデータの個数をp, 0以上k以下のキーを持つデータの個数の累計をqとし，分布表に従ってキーkを持つデータを結果の配列の(q-p)番目から(q-1)番目に並べれば，昇順に並べられたデータを得ることができる．

表1　分布表

キー（k）	個数（p）	累計（q）
0	0	0
1	1	1
2	0	1
3	2	3
4	1	4
5	1	5

以下の関数 dist_count_sort は，上述の分布数え上げソートのアルゴリズムをC言語で書いたものである．関数 dist_count_sort は，データの入った配列aと配列aの要素数nを受け取り，aの要素を分布数え上げソートによって昇順に並べ替え，結果を配列bに代入する．このプログラムに関して以下の問いに答えなさい．

(2-1) データの並びを 1,4,2,7,8,2 とした場合のキーの分布表を書きなさい．キーの範囲は0以上10以下とする．

(2-2) 関数 dist_count_sort 中の空欄（ア）と（イ）にC言語の式を1つずつ埋めなさい．

(2-3) 関数 dist_count_sort は安定な整列を実現できる．安定な整列とはどのような整列であるか説明しなさい．また，(2-1) で用いたデータの並び 1,4,2,7,8,2 において，3番目の「2」と6番目の「2」が区別されるものと

次ページに続く

して，配列 b に得られる結果が安定であることを説明しなさい．

```
#define   M    10
void dist_count_sort(int a[], int b[], int n)
{
    int count[M+1];
    int i;

    for (i = 0; i <= M; i++)   /* キーの出現回数を初期化する */
        count[i] = 0;

    for (i = 0; i < n; i++)   /* キーの出現回数を調べる   */
        (ア)

    for (i = 0; i < M; i++)    /* 数え上げたキーの累積度数分布を求める */
        (イ)

    /* 度数分布に従ってデータを配列 a から配列 b にコピーする */
    for (i = n-1; i >= 0; i--)
        b[--count[a[i]]] = a[i];
}
```

問題４　情報基礎２

本問題ではバーコードの規格の一つであるJapanese Article Number (JAN) コードを取り上げる．バーコードは，複数の白色のスペースと黒色のバーの組み合わせで構成される．0を白色，1を黒色で表すと，例えば7ビットのデータ「0001101」は図1のようになる．図2に示すバーコードは右から，3ビットで表されるライトガードバー，7ビットで表されるチェックディジット，7ビットを1文字として3文字分のデータから成る右側データキャラクタ，5ビットで表されるセンターバー，7ビットを1文字として4文字分のデータから成る左側データキャラクタ，3ビットで表されるレフトガードバーで構成される．8桁のJANコードは，4文字の左側データキャラクタ，3文字の右側データキャラクタ，1文字のチェックディジットを用いて，合計8文字で構成される．また表1に示すように，ライトガードバー，センターバー，レフトガードバーはそれぞれ固定のパターンで表される．

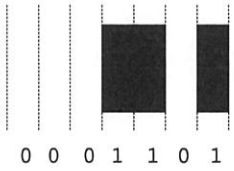

図1: 7ビットのデータのバーコードの例

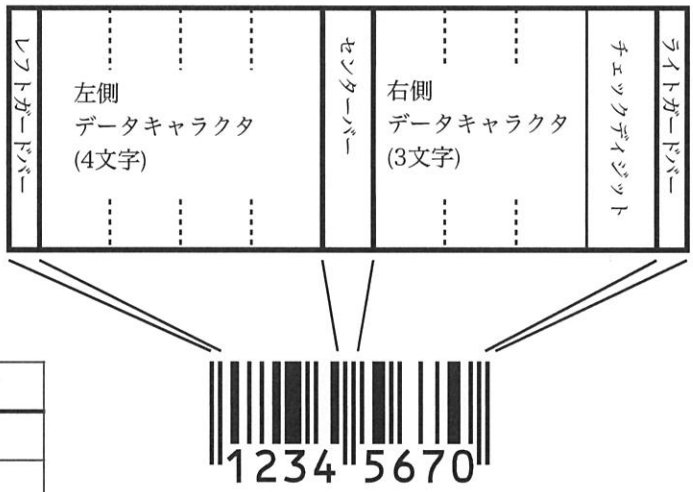

図2: バーコードシンボルとその構成

表1: バーのキャラクタ

バー	キャラクタ
ライトガードバー	101
センターバー	01010
レフトガードバー	101

8桁のJANコードでは0～9の数字が使用され，表2に示すように各数字は1文字あたり7ビットで表される．ただし同じ数字であっても，右側と左側では異なるデータキャラクタを用いる．例えば図1に示すバーコードの場合，左側データキャラクタの「０」を表す．

次ページに続く

表2: データキャラクタとJANコードの数値の対応

JANコードの数値 (10進数)	左側データキャラクタ(1文字分)	右側データキャラクタおよび チェックディジット(1文字分)
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

JANコードからバーコードを生成するプログラムと，読み取ったバーコードからJANコードへ変換するプログラムを作成することを想定し，以下の問いに答えなさい．ただしプログラミング言語はC言語を用いるものとする．

(1) 図3のバーコードが示す8桁のJANコードの右から3桁目と8桁目を答えなさい．ただし，区切りを分かりやすくするために1ビット毎に点線を付加した．

図3: バーコード

(2) 表2に示す右側データキャラクタを，以下に示すグローバル定数として宣言する．1文字は7ビットで表されるのでunsigned char型とする．例えばプログラム中でright_characters[5]を参照すると，JANコードの数値「5」に相当するデータキャラクタが得られる．空欄(ア)と(イ)を埋めなさい．

```
const unsigned char right_characters[10] = {
    0x[    ], 0x[    ], 0x[ (ア) ], 0x[    ], 0x[    ],
    0x[    ], 0x[    ], 0x[ (イ) ], 0x[    ], 0x[    ]};
```

次ページに続く

(3) 8桁のJANコードを表すバーコードは，67個のビットデータで構成されるため，長さ9のunsigned char型の配列barcode[]に格納される．格納の順番は，最下位ビットにバーコードの最も右側のビットを対応させる．配列とビットデータは図4に示すように格納される．barcode[]の最下位ビットからnビット目（ただしn=0, … , 66）の値を「1」に変更する関数set_bit()，「0」に変更する関数clear_bit()，nビット目の値を取得する関数get_bit()を以下のように作成する．空欄(ウ)と(エ)を埋めなさい．

```
void set_bit(unsigned char *barcode, int n) {
  *(barcode+(  (ウ)  )) |= (unsigned char)(  (エ)  );
}

void clear_bit(unsigned char *barcode, int n) {
  *(barcode+(  (ウ)  )) &= (unsigned char)(~(  (エ)  ));
}

int get_bit(const unsigned char *barcode, int n) {
  if(*(barcode+(  (ウ)  )) & (unsigned char)(  (エ)  )) {
    return 1;
  } else {
    return 0;
  }
}
```

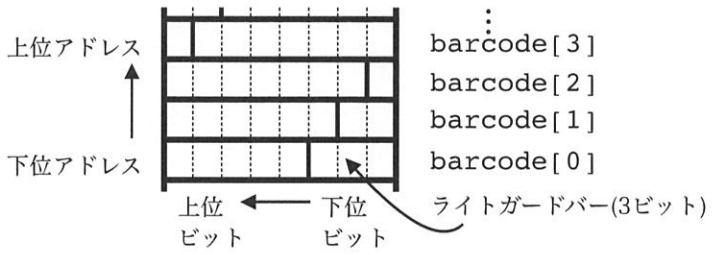

図4: 配列とビットデータ

(4) 関数recognize_right_character()は，バーコードを表す67個のビットデータを入力として受け取り，8桁のJANコードの右からm+1桁目（ただしm=0, … , 3）を出力とする関数である．67個のビットデータは問(3)と同様にbarcode[]に格納され，最下位ビットにバーコードの最も右側のビットを対応させる．JANコードは配列unsigned int jancode[]に格納され，JANコードが「12345670」の場合，図5のようになる．ここでは，JANコードの右からm+1桁目に対応する7ビット分のデータがいずれかの右側データキャラクタと一致すれば，それに対応するJANコードの数値をjancode[]の適切な場所に格納し，いずれとも一致しなければ「偽」を関数の戻り値として返す．空欄(オ)〜(キ)を埋めなさい．なお問(2)と問(3)で作成した配列や関数を用いてよい．

次ページに続く

```
int recognize_right_character(int m, unsigned char *barcode,
unsigned int *jancode)
{
  int i;
  int offset;
  unsigned char number=0;

  offset = [(オ)                    ];
  for (i=0; i<7; i++) {
    if (get_bit(barcode, offset)) {
       [(カ)                    ];
    }
    offset++;
  }

  for (i=0; i<10; i++) {
    if (number == right_characters[i]) {
       * [(キ)                    ];
       return 1;
    }
  }
  return 0;
}
```

jancode[7]	1
jancode[6]	2
jancode[5]	3
jancode[4]	4
jancode[3]	5
jancode[2]	6
jancode[1]	7
jancode[0]	0

図5: JANコードの格納順

(5) バーコードから7ビット1文字分のデータキャラクタを読み取る際，「パリティチェック」と呼ばれる読み取り誤り検出が行われる．右側データキャラクタは偶数パリティ，左側データキャラクタは奇数パリティである．その関数parity_check()を以下に示す．引数としてodd_evenには奇数パリティの場合は1を，偶数パリティの場合は0を指定し，*numberにはデータキャラクタを表すビット配列の先頭アドレスが指定される．この関数の戻り値のそれぞれの値が何を表しているかを説明しなさい．また，検出できない読み取り誤りにはどのようなものがあるか，具体例とともに示しなさい．なお関数get_bit()は問(3)で示したものである．

```
int parity_check(int odd_even, unsigned char *number)
{
  int i;
  int b=0;

  for (i=0; i<7; i++) {
    if (get_bit(number, i)) {
      b++;
    }
  }
  if (b % 2 == 0) {
    if (odd_even == 0) return 1;
    else               return 0;
  } else {
    if (odd_even == 0) return 0;
    else               return 1;
  }
}
```