知能システム学Ｉ

次の3問すべてを解答せよ．問題毎に別の答案用紙を用いること．答案用紙の追加は認めない．

1

以下の問い 1) ～ 3) に答えよ．

1) 次の線形微分方程式の一般解を求めよ．

$$x^2\frac{dy}{dx} - 2xy = 3$$

2) 次の非線形微分方程式の一般解を求めよ．ただし，$z = \dfrac{1}{y}$ の変換を用いてよい．

$$\frac{dy}{dx} + 2\frac{y}{x} = x^2y^2\sin x$$

3) 次の微分方程式は，適当な変換によって，線形微分方程式になることを示せ．

$$\frac{dy}{dx} + P(x)y = Q(x)y^n$$

ただし，n は2以上の自然数である．

2

以下の問い 1)～4) に答えよ．

1) 以下の行列

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & t \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

に対して，$B = S^T A S$ とおく．また，t は実数とし，S^T は行列 S の転置を表す．次の小問 a)～c) に答えよ．

a) B を計算せよ．

b) B が正定値行列となるための t に関する条件を求めよ．

c) S と B の階数を求めよ．

2) 2 次方程式 $3x^2 - 2\sqrt{2}xy + 2y^2 = 1$ の標準形を求め，解の軌跡を表す楕円を描け．ここで，楕円の長軸と短軸の傾きと長さを図中に記入すること．

3) 4 次元実数空間上のベクトル a_1, a_2, a_3, a_4 を

$$a_1 = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 4 \end{bmatrix}, \quad a_2 = \begin{bmatrix} -1 \\ 1 \\ 3 \\ -3 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 0 \\ 1 \\ -5 \\ -2 \end{bmatrix}, \quad a_4 = \begin{bmatrix} -1 \\ -9 \\ -1 \\ -4 \end{bmatrix}$$

とする．a_1, a_2 の張る部分空間を W_1 とし，a_3, a_4 の張る部分空間を W_2 とするとき，共通部分 $W_1 \cap W_2$ の基底ベクトルを求めよ．

4) 2 以上の自然数 n に対して次の正方行列を考える．

$$A_n = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}$$

この行列 A_n の行列式 $|A_n|$ が

$$|A_n| = \prod_{1 \le i < j \le n} (x_j - x_i)$$

となることを示せ．

3

以下の文字列探索に関する問い 1), 2) に答えよ．ただし，プログラミング言語は C 言語を用いている．各問いで対象とする文字列はアスキー文字のみ含んでいるものとする．

1) プログラム 1 は，力まかせのアルゴリズム（brute force algorithm）による文字列探索のプログラムである．これは，ある特定の文字列（テキスト）の先頭に，探索で見つけたい文字列（パターン）を重ね合わせ，1 文字ずつそれらが一致するかどうかを確かめる．一致しない文字があれば，パターンを 1 文字分だけ後ろにずらして，再びパターンとテキストを比較する．関数 brute_force_search の引数は，テキスト（text）とその長さ（text_len），パターン（pattern）. とその長さ（pat_len）である．探索が成功した場合はパターンが見つかった位置（テキスト中でパターンの先頭に対応する位置. テキスト中の先頭文字を 1 文字目とする）を，探索が失敗した場合は-1 を返り値とする．以下の小問 a) ～ c) に答えよ．

a) プログラム 1 中の空欄 [1] と [2] を埋めよ．なお，空欄 [1] は複数の文で構成してもよい．セミコロン抜けに注意せよ．

b) テキストの長さを n，パターンの長さを m としたとき，このアルゴリズムの平均時間計算量を n と m を用いてオーダー表記で表せ．

c) 入力されるテキストやパターンが一般的な自然言語で書かれている場合の，実質的な平均時間計算量を，小問 b) で定義した n と m を用いて，オーダー表記で表せ．

プログラム 1

```
int brute_force_search(char text[], int text_len, char pattern[], int pat_len)
{
    int i, j;
    i = 0; j = 0;
    while (i < text_len && j < pat_len) {
        if (text[i] == pattern[j]) {
            i++; j++;
        } else {
            [1]
        }
    }
    return (j == pat_len) ? [2];
}
```

———次ページに続く———

2)　プログラム 2 は，Boyer-Moore のアルゴリズムによる文字列探索のプログラムである．これは，ある特定の文字列（テキスト）の先頭に探索で見つけたい文字列（パターン）を重ね合わせ，パターンの末尾から先頭に向かって順番に文字を比較していく．パターンとテキストに不一致が見つかれば，不一致の原因になった文字に応じてパターンをずらす分量を決める．関数 boyer_moore_search の引数は，テキスト（text）とその長さ（text_len），パターン（pattern）とその長さ（pat_len）である．探索が成功した場合はパターンが見つかった位置（テキスト中でパターンの先頭に対応する位置．テキスト中の先頭文字を 1 文字目とする）を，探索が失敗した場合は-1 を返り値とする．なお，プログラム中の関数 max は，二つの整数を引数として持ち，そのうち大きい方の値（二つとも同じ値であれば，その値）を返り値として返すものである．以下の小問 a），b）に答えよ．

a) プログラム 2 中の空欄 [1] ～ [4] を埋めよ．

b) テキストの長さを n，パターンの長さを m と表す．入力されるテキストやパターンが一般的な自然言語で書かれている場合の，実質的な平均時間計算量をオーダー表記で表せ．

プログラム 2

```
int boyer_moore_search(char text[], int text_len, char pattern[], int pat_len)
{
        int skip[128];
        int i, j;
        for (i = 0; i < 128; i++)
                skip[i] = [ 1 ];
        for (i =0; i < pat_len - 1; i++)
                skip[pattern[i]] = [ 2 ];

        i = [ 3 ];
        while (i < text_len) {
                j = pat_len - 1;
                while (text[i] == pattern[j]) {
                        if (j == 0)
                                return i;
                        i--; j--;
                }
                i = i + max(skip[text[i]], [ 4 ]);
        }
        return -1;
}
```