

令和7年度 豊橋技術科学大学第3年次入学者選抜学力検査問題

専 門 科 目 （ 3 : 情 報 ・ 知 能 工 学 ）

注 意 事 項

- 1 試験開始の合図まで、この問題冊子と解答用紙を開いてはいけません。
- 2 問題冊子の枚数は表紙、草稿用紙を含めて8枚です。
- 3 問題冊子とは別に解答用紙が4枚あります。解答は用紙の裏面にまわってはいけません。
- 4 問題は3問あります。全問解答してください。
- 5 試験開始の合図の後すぐに、すべての解答用紙の所定の箇所に受験番号を記入してください。
- 6 解答は必ず各問題別の解答用紙の所定の欄に記入してください。
- 7 落丁、乱丁、印刷不鮮明の箇所などがあれば、ただちに申し出てください。
- 8 問題冊子の余白は草稿用として使用しても構いません。
- 9 試験終了時刻まで退出してはいけません。
- 10 問題冊子は持ち帰ってください。

(草稿用紙)

[1] 関数 $f(x)$, $g(x)$ について, 以下の空欄に入る値を答えよ。

(1) 関数 $f(x)$ は微分方程式 $\frac{d^2 f(x)}{dx^2} = ax^2 + b$ (a, b は定数) の解であるとする。

$$\left. \frac{d^2 f(x)}{dx^2} \right|_{x=2} = \frac{1}{2}, \quad \left. \frac{d^2 f(x)}{dx^2} \right|_{x=-1} = -\frac{1}{4} \text{ のとき, } a = \boxed{\text{ア}}, \quad b = \boxed{\text{イ}} \text{ である。}$$

さらに $f(x)$ を偶関数とし, $f(2) = -2$ であるとする, 微分方程式の解は

$$f(x) = \boxed{\text{ウ}} x^4 + \boxed{\text{エ}} x^2 + \boxed{\text{オ}} \text{ である。}$$

(2) 関数 $g(x)$ を $g(x) = \frac{1}{x^2} \left\{ -1 + \cos\left(\frac{\pi}{2}x\right) \right\}$ (ただし $x \neq 0$) と定義する。

$\cos\left(\frac{\pi}{2}x\right)$ を $x=0$ のまわりでテイラー展開すると,

$$\cos\left(\frac{\pi}{2}x\right) = 1 - \boxed{\text{カ}} x^2 + \boxed{\text{キ}} x^4 - \dots \text{ となるので,}$$

$$\lim_{x \rightarrow +0} g(x) = \boxed{\text{ク}} \text{ である。}$$

(3) $-1 \leq \cos\left(\frac{\pi}{2}x\right) \leq 1$ より関数 $g(x)$ の最大値は $\boxed{\text{ケ}}$ である。 $0 < x < 5$ の範囲

で最大値を与える x は $x = \boxed{\text{コ}}$ である。また, $x = \boxed{\text{コ}}$ における

$f(x)$ の値は $\boxed{\text{サ}}$ である。

- [2] 1次元配列に格納された数字を小さい方から大きい方へ並べ替える昇順ソートプログラムについて検討する。本問で示すC言語(ANSI C)で書かれたプログラムはいずれもソートを実行するプログラムであり、main関数から配列(array)と配列の長さ(size)を引数として、関数A_sortおよび関数B_sortをそれぞれ呼び出すことで実行される。

以下の文章およびプログラム中にある空欄 , に入る適当なもの1つを本問の最後に示す選択肢から選び、その番号①～⑥で答えよ。同じ番号を何度使用してもよい。また、空欄 ～ には数値を、空欄 ～ には配列arrayのすべての要素をそれぞれ答えよ。

- (1) プログラム1に関数A_sortおよび配列の要素表示用の関数print_arrayを示す。A_sortは、配列の先頭要素から順に隣り合う要素と比較し昇順でなければ交換する。配列の先頭から開始した比較・交換が配列の最後まで進むと、配列の末尾要素は最大値となる。このため、次に配列の先頭から開始する比較・交換は前回より1つ手前までとする。以上の操作を繰り返すことによりソートが完了する。array[6]={2, 6, 4, 3, 1, 5};としてA_sortを呼び出し実行すると、(1.17)行目における関数print_arrayの1回目の呼び出しで表示されるarrayは となる。A_sortの実行が終了するまでに、(1.12)行目の比較は 回行われ、(1.14)行目の交換は 回行われる。
- (2) プログラム2に関数B_sort、関数B_sort_rec、関数print_arrayを示す。B_sort_recでは、初めに基準値(pivot)を配列の中央位置の要素の値と定義する。配列の先頭側から基準値以上の値が出現する要素まで位置LOを進めると同時に、末尾側から基準値以下の値が出現する要素まで位置HIを進め、それら2つの位置が決定した後、位置LOと位置HIの要素を交換する。両側から進めた要素の位置LOとHIが一致もしくはLOよりHIの位置が先頭側にあればループを抜ける。交換後の基準値以下の要素のグループと基準値以上の要素のグループのそれぞれに対して、要素数が2以上であれば再帰呼び出しを行う。array[6]={2, 6, 4, 3, 1, 5};としてB_sortを呼び出し実行すると、(2.13)行目の1回目の実行で決まるpivotの値は となる。(2.25)行目における関数print_arrayの1回目の呼び出しで表示されるarrayは となり、2回目の呼び出しで表示されるarrayは となる。B_sortの実行が終了するまでに(2.21)行目の交換は 回行われる。また、(2.28)行目および(2.32)行目でB_sort_recが呼び出されるのは合計 回である。

```
(1.01)#define SWAP(a,b) { int tmp = a; a = b; b = tmp; }
(1.02)void A_sort(int array[], int size);
(1.03)void print_array(const int array[], int size);
(1.04)
(1.05)void A_sort(int array[], int size)
(1.06){
(1.07)    int i, j;
(1.08)    for (i = 0; i < size; i++)
(1.09)    {
(1.10)        for (j = 0; j < ア; j++)
(1.11)        {
(1.12)            if (array[j+1] < array[j])
(1.13)            {
(1.14)                SWAP(array[j+1], array[j]);
(1.15)            }
(1.16)        }
(1.17)        print_array(array, size);
(1.18)    }
(1.19)}
(1.20)
(1.21)void print_array(const int array[], int size)
(1.22){
(1.23)    int i;
(1.24)    for (i = 0; i < size; i++)
(1.25)    {
(1.26)        printf("%d", array[i]);
(1.27)        if(i != size-1) printf(", ");
(1.28)    }
(1.29)    printf("¥n");
(1.30)}
```

プログラム 1

```

(2.01)#define SWAP(a,b) { int tmp = a; a = b; b = tmp; }
(2.02)void B_sort(int array[], int size);
(2.03)void B_sort_rec(int array[], int begin, int end, int size);
(2.04)void print_array(const int array[], int size);
(2.05)
(2.06)void B_sort(int array[], int size)
(2.07){
(2.08)    B_sort_rec(array, 0, , size);
(2.09)}
(2.10)
(2.11)void B_sort_rec(int array[], int begin, int end, int size)
(2.12){
(2.13)    int pivot = array[(begin+end)/2];
(2.14)    int i = begin; /* iは問題文中の位置LO */
(2.15)    int j = end; /* jは問題文中の位置HI */
(2.16)    while(1)
(2.17)    {
(2.18)        while(array[i] < pivot){i++;}
(2.19)        while(array[j] > pivot){j--;}
(2.20)        if (i >= j ){ break; }
(2.21)        SWAP(array[i], array[j]);
(2.22)        i++;
(2.23)        j--;
(2.24)    }
(2.25)    print_array(array, size);
(2.26)    if (i-begin >= 2 )
(2.27)    {
(2.28)        B_sort_rec(array, begin, i-1, size);
(2.29)    }
(2.30)    if (end-j >= 2 )
(2.31)    {
(2.32)        B_sort_rec(array, j+1, end, size);
(2.33)    }
(2.34)}
(2.35)
(2.36)void print_array(const int array[], int size)
(2.37){
(2.38)    int i;
(2.39)    for(i = 0; i < size; i++)
(2.40)    {
(2.41)        printf("%d", array[i]);
(2.42)        if(i != size-1) printf(", ");
(2.43)    }
(2.44)    printf("¥n");
(2.45)}

```

プログラム2

選択肢

① size	② size-1	③ size+1	④ size-i
⑤ size-1-i	⑥ size+1-i		

[3] 2進数3桁の数 $b_2b_1b_0$ に対応する10進数1桁の数0から7までを7セグメントディスプレイにて表示するための論理回路について以下の問いに答えよ。7セグメントディスプレイとは、図1のように L_0 から L_6 の7つのセグメントから構成されるLEDを点灯または消灯することにより図2に示すような数字を表すものである。なお、「 \cdot 」は論理積、「 $+$ 」は論理和、「 \bar{X} 」は変数 X の否定を表すものとし、真理値表およびカルノー図においては真の値を1、偽の値を0で表すものとする。

- (1) 2進数の入力 $b_2b_1b_0$ に対応するLEDのセグメント L_0 から L_6 の状態を示す論理関数 F_0 から F_6 の真理値表を解答用紙に完成させよ。なお、LEDにおいては点灯状態を'1'、消灯状態を'0'とする。
- (2) L_2 の状態に対応する論理関数 $F_2(b_2, b_1, b_0)$ を標準和積形で表現せよ。
- (3) L_5 の状態に対応する論理関数 $F_5(b_2, b_1, b_0)$ を標準積和形で表現せよ。
- (4) 論理関数 F_5 の最小積和形の論理式を、カルノー図を用いて求めよ。解答欄にはカルノー図も示すこと。なお、最小積和形とは、複数個の論理積項が論理和で結ばれている論理式において、論理積と論理和の総数が最小になる形式とする。
- (5) NANDゲートは完全系であり、NANDゲートのみを用いて任意の論理回路を構成できることが知られている。2つの入力 A, B に対してOR演算を行う論理回路を2入力NANDゲートを用いて構成し、その回路図を解答用紙に完成させよ。なお、2入力NANDゲートは図3のように表現されるとする。
- (6) 問(2)にて求めた標準和積形で示される論理関数 F_2 を、最小個数の2入力NANDゲートのみを用いて構成し、その回路図を解答用紙に完成させよ。

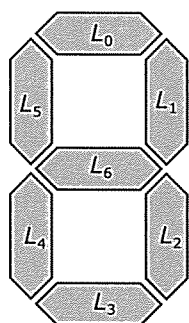


図1 7セグメントディスプレイ

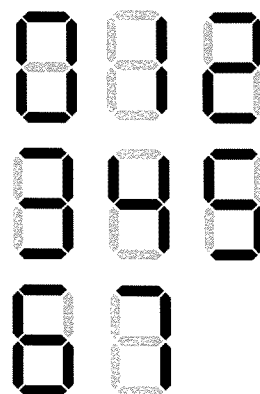


図2 0から7の数字の表示



図3 2入力NANDゲート