名古屋工業大学

２０２０年度編入学者・転入学者選抜学力検査[問題]

－ 専 門 試 験 －

(情報工学科)

試 験 日 時　２０１９年6月２１日（金）

１０：００～１２：００

●解答上の注意

(1) 解答の際，解答用紙のホチキス止めを外してください。
(2) 配布物は，問題冊子１冊，解答用紙３枚，計算用紙１枚です。
(3) 解答は各問題番号に対応する解答用紙に解答してください。
(4) 解答が解答用紙表面に書ききれない場合は，裏面に続いてもよいが，その場合は表面の下側が裏面の上側になるようにし，上側２／３のスペースに解答を収めてください。
(5) 電卓は使用できません。
(6) 試験終了後は問題用紙と計算用紙を持ち帰ってください。

問題 1　設問すべてについて解答すること。ただし，回路図を示す場合には，記号として図 1-1 に示す論理記号を用いること。また，A の否定を $\overline{\mathrm{A}}$ で表す。

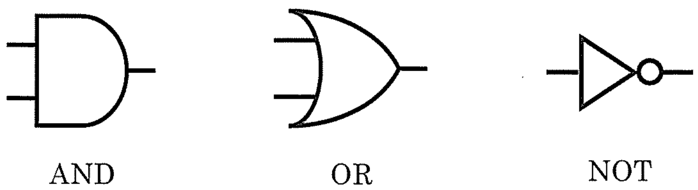


図 1-1：論理記号

I　2 の補数表現をした符号付き 2 進整数について，以下の(1)～(3)の問いに答えよ。ただし，8 ビットで表記するものとする。

(1) 以下の 4 つの 2 進整数の 2 の補数をそれぞれ求めよ。

①　00000001　　②　01101001　　③　10000001　　④　10010111

(2) 以下の計算をせよ。

01101001 + 10010111

(3) 一般に，2 つの 2 進整数の差を求めるには，減数（引く数）の 2 の補数を被減数（引かれる数）に加算し，桁あふれを捨てる。このようにするとなぜ差が求まるのかを説明せよ。

II　JK 型フリップフロップは，2 つの入力 J と K，内部状態 Q を持ち，次状態 Q_{next} とその否定を出力するフリップフロップである。 $J = K = 0$ の時に $Q_{next} = Q$，　$J = 1, K = 0$ の時に $Q_{next} = 1$，　$J = 0, K = 1$ の時に $Q_{next} = 0$，　$J = K = 1$ の時に $Q_{next} = \overline{Q}$ となる。この時，以下の(1)～(2)の問いに答えよ。

(1) J, K, Q を入力，Q_{next} を出力として，真理値表とカルノー(Karnaugh)の図を示せ。

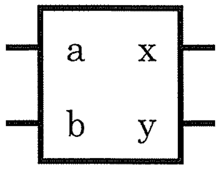
(2) Q_{next} を J, K, Q の NOT-AND-OR 形式の論理式で表せ。できるだけ簡単化すること。

Ⅲ　以下の(1)～(3)の問いに答えよ。

(1) 右の真理値表で，a, b を入力，x, y を出力とする。x と y をそれぞれ a, b の最小項の論理和である積和標準形（主加法標準形）で記し，a, b を入力とし x, y を出力とする回路図を示せ。

a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(2) (1)の回路を 2 入力 2 出力の回路として右図で表現することにする。この時，図 1-2 で表される回路の真理値表を，入力を A, B, C，出力を X, Y として記せ。



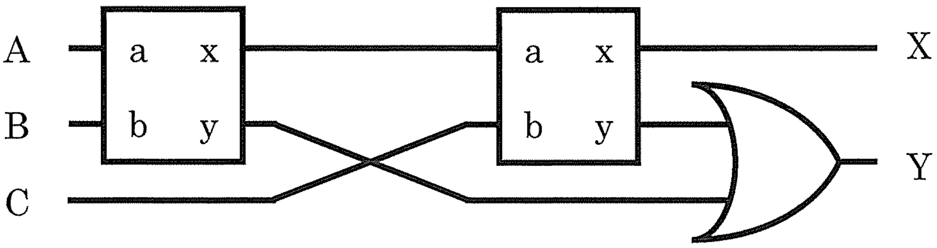
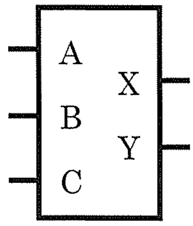


図 1-2

(3) 図 1-2 の回路を 3 入力 2 出力の回路として右図で表現することにする。この時，図 1-3 で表される回路について，以下の①～②の問いに答えよ。



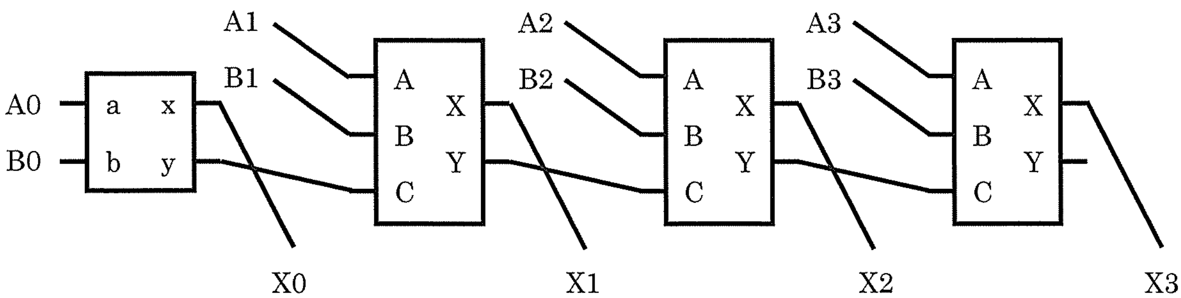


図 1-3

① A0 = 0, A1 = 1, A2 = 1, A3 = 0, B0 = 1, B1 = 1, B2 = 0, B3 = 0 の時の，X0, X1, X2, X3 の値を求めよ。

② この回路は入力に対し何を出力するものか説明せよ。

問題２　設問すべてについて解答すること。ただし，プログラム中の□で隠された部分の大きさと，本来の記述の長さとは無関係である。また，必要な#include 文は記述されているとみなせ。

I　図 2-1 のプログラム 1 は，文字列 chars に含まれる文字を使ってランダムに生成した文字列が，回文（すなわち，左から読んでも右から読んでも同じ文字列）になっている確率の実測値とその理論値を求める C 言語のプログラムである。このとき，次の(1)〜(5)の問いについて答えよ。ただし，gen_len は生成する文字列の長さ，strlen(s)は与えられた文字列 s の長さを返す関数，random(n)は 0 以上 n 未満の整数の一様乱数を返す関数とする。

```
char chars[] = "abcdefg";
int gen_len = 5;

void gen_random_str(char* str) {
  int i, char_idx;
  for (i = 0; i < gen_len; i++) {
    char_idx = random(strlen(chars));
    str[i] = chars[char_idx];
  }
  str[gen_len] = '¥0';
}

int is_kaibun(char* str) {
  int i, j;
  int n = strlen(str);
  for (i = 0; i < n / 2; i++) {
    j = [    ア    ] ;
    if (str[i] != str[j]) {
      return [    イ    ] ;
    }
  }
  return [    ウ    ] ;
}
```

```
float esti_prob() {
  [    エ    ]
}

int main() {
  int i;
  int trial = 500;
  int count = 0;
  char str[gen_len + 1];
  for (i = 0; i < trial; i++) {
    gen_random_str(str);
    if (is_kaibun(str)) {
      count++;
    }
  }
  printf("実測値: %f¥n", (float)count / trial);
  printf("理論値: %f¥n", esti_prob());
  return 0;
}
```

図 2-1: プログラム 1

(1) 文字列 str が回文のとき関数 is_kaibun(str)が 1 を返し，そうでないとき 0 を返せるよう，空欄ア〜ウに入る適切な記述を答えよ。

(2) chars 内に同じ文字が複数含まれないとき，関数 gen_random_str で生成される文字列が回文になる確率の理論値pは，以下の式で与えられる。ただし，式中のclは文字列 chars の長さ，glは gen_len の値を表すものとする。

$$p = \begin{cases} cl^{-\frac{gl}{2}} & (gl\text{が偶数のとき}) \\ cl^{(\frac{1-gl}{2})} & (gl\text{が奇数のとき}) \end{cases}$$

このとき，gen_len の任意の値に対して関数 esti_prob()が回文になる確率の理論値を返せるように，空欄エに入る適切な記述を答えよ。ただし，x の y 乗を返す関数 pow(x, y)を使うこと。

(3) 図 2-2 のプログラム 2 の関数 is_kaibun_rec(str, i)は，プログラム 1 の is_kaibun(str)を再帰関数として書き直したものである。このとき，空欄オ～キに入る適切な記述を答えよ。ただし，is_kaibun_rec 中の空欄アにはプログラム 1 の空欄アと同じ記述が入る。

(4) プログラム 2 の関数 gen_ordered_str(str, k)は，プログラム 1 で求めた理論値を検算するために，与えられた整数値 k を文字列に変換して変数 str に格納する関数である。たとえば，gen_ordered_str(str, 48)を呼び出すと "ggaaa" という文字列が str に格納される。これにより main 関数では，プログラム 1 の gen_random_str で生成可能な全文字列について調べ，総当たりによる検算ができる。このとき，空欄クに入る適切な記述を答えよ。

(5) gen_ordered_str(str, 8)を呼び出したときに str に格納される文字列を答えよ。

```
char chars[] = "abcdefg";
int gen_len = 5;

int is_kaibun_rec(char* str, int i) {
    int n = strlen(str);
    int j = [   ア   ] ;
    if (j > i) {
        if (str[i] == str[j]) {
            return [   オ   ] ;
        } else {
            return 0;
        }
    } else {
        [   カ   ] ;
    }
}
```

```
void gen_ordered_str(char* str, int k) {
    int i, char_idx;
    int n = strlen(chars);
    for (i = 0; i < gen_len; i++) {
        char_idx = k % n;
        str[i] = chars[char_idx];
        k = [   ク   ] ;
    }
    str[gen_len] = '¥0';
}

int main() {
    int i;
    int trial = pow(strlen(chars), gen_len);
    int count = 0;
    char str[gen_len + 1];
    for (i = 0; i < trial; i++) {
        gen_ordered_str(str, i);
        if (is_kaibun_rec(str, [   キ   ] )) {
            count++;
        }
    }
    printf("理論値: %f¥n", (float)count / trial);
    return 0;
}
```

図 2-2: プログラム 2

II 図 2-3 のプログラム 3 は，非負の整数を格納するキューを実装した C 言語のプログラムである。関数 enqueue(input)で末尾に要素 input を追加し，関数 dequeue()で先頭から要素を取り出す。要素が無い箇所には，-1 を代入するものとする。このとき，次の(1)～(3)の問いについて答えよ。

```
#define MAX_SIZE 32

int queue[MAX_SIZE];
int last;

int enqueue(int input) {
  if (last < MAX_SIZE) {
    queue[last] = input;
    last ++;
    return 0;
  } else {
    printf("要素数が上限に達しました¥n");
    return -1;
  }
}

int dequeue() {
  int output = queue[  ケ  ];
  int i;
  if (output >= 0) {
    for(i = 1; i < last; i++) {
      queue[i-1] = queue[i];
    }
    queue[  コ  ] = -1;
    last--;
    return output;
  } else {
    printf("要素がありません¥n");
    return -1;
  }
}
```

```
void init_queue() {
  int i;
  for (i = 0; i < MAX_SIZE; i++) {
    queue[i] = -1;
  }
  last = 0;
}

void show_queue() {
  int i;
  printf("queue: ");
  for (i = 0; i < MAX_SIZE && queue[i] >= 0; i++) {
    if (i > 0) {
      printf(",");
    }
    printf("%d", queue[i]);
  }
  printf("¥n");
}

int main() {
  int i;
  int d;
  init_queue();
  for (i = 0; i < 5; i++) {
    enqueue(2 * i + 1);
  }
  d = dequeue();
  printf("d: %d¥n", d);
  show_queue();
  printf("size: %d¥n", last);
  return 0;
}
```

図 2-3: プログラム 3

(1) キューのようなデータの出し入れ方式を FIFO と呼ぶが，この 4 文字“F”，“I”，“F”，“O”はそれぞれある単語の頭文字である。それぞれの単語を答えよ。

(2) 空欄ケ，コに入る適切な記述を答えよ。

(3) 図 2-4 に，プログラム 3 の実行後の標準出力への出力結果を示す。空欄サ～スに入る適切な出力結果を答えよ。

d: サ
queue: シ
size: ス

図 2-4: プログラム 3 による出力結果

問題3　設問すべてについて解答すること。導出過程も簡潔に示すこと。ただし，解答においては最も簡約化した形で示すこと。ここで簡約化とは，分数に関しては既約形，対数に関しては最も簡単な形（例：$\log_2 6 = 1 + \log_2 3$）に変形することを指す。また，$0 \log_2 0 = 0$とする。

I　情報源アルファベットを$S = \{a, b, c, d, e\}$，符号アルファベットを$T = \{0,1\}$とする。S上の確率変数Xが従う確率分布Pと，Sに対する可変長符号Cが以下の表のように与えられている。

X	a	b	c	d	e
$P(X)$	3/8	1/4	1/8	1/8	1/8
$C(X)$	00	01	101	110	111

このとき，次の(1)～(4)の問いについて答えよ。

(1)　Xのエントロピー$H(X)$を求めよ。

(2)　符号Cの平均符号長$L(C)$を求めよ。

(3)　符号Cがクラフトの不等式を満たすことを示せ。

(4)　符号Cの平均符号長よりも短い平均符号長をもつ符号としてハフマン符号C_Hを考える。符号C_Hの平均符号長$L(C_H)$を求めよ。

II　図 3-1 に示した通信路において，送信記号をX，受信記号をYとする。また，送信記号Xの生起確率を$P(X=1)=a$，$P(X=0)=1-a$ とする。このとき，次の(1)～(4)の問いについて答えよ。ただし，$0 \leq a \leq 1$，$0 \leq p \leq 1$とし，必要であれば二値エントロピー関数

$$h(a) = -a\log_2 a - (1-a)\log_2(1-a)$$

を用いること。

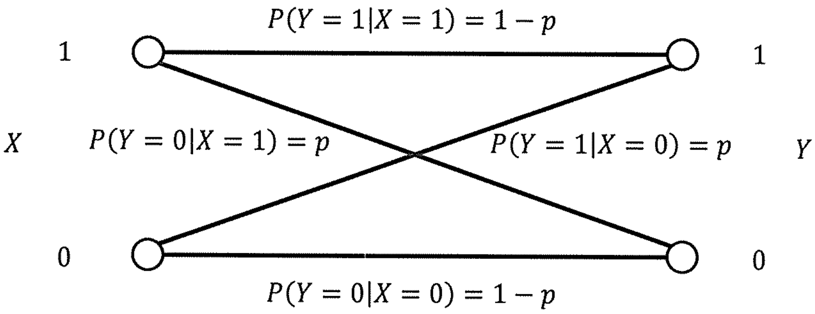


図 3-1：通信路線図

(1) 受信記号Yが 1，0 となる確率$P(Y=1)$，$P(Y=0)$，および条件付確率$P(X=1|Y=1)$をa，pを用いて表せ。

(2) 相互情報量$I(X;Y)$を求めよ。

(3) この通信路の通信路容量C，およびCを達成するaの値を求めよ。

(4) 通信路容量Cをpの関数とみたとき，その最大値，最小値をそれぞれ求めよ。