

情報基礎（90分）

〔注意事項〕

1. 監督者の指示があるまで、この問題用紙と解答用紙を開いてはいけません。
2. 問題は、7ページからなっています。また、解答用紙は3枚、下書用紙は1枚あります。監督者から解答開始の合図があったら、問題用紙、解答用紙、下書用紙を確認し、落丁・乱丁および印刷の不鮮明な箇所などがあれば、手をあげて監督者に知らせなさい。
3. 解答用紙には、受験番号を記入する欄がそれぞれ2箇所ずつあります。監督者の指示に従って、すべての解答用紙（合計3枚）の受験番号欄（合計6箇所）に受験番号を必ず記入しなさい。
4. この問題用紙の白紙と余白は、適宜下書きに使用してよろしい。
5. 解答は、必ず解答用紙の指定された場所（問題番号や設問の番号・記号などが対応する解答欄の中）に記入しなさい。なお、指定された場所以外や、裏面への解答は採点対象外です。また、解答や受験番号が判読不能の場合にも、採点対象外になります。
6. 解答用紙は持ち帰ってはいけません。
7. この問題用紙と下書用紙は、持ち帰りなさい。

I

(1) 次に示すプログラムは、整数の 2 進数ビットパターンを用いてファイルのアクセス権限を管理するものである。空欄 ①～③を埋めて、プログラムを完成させなさい。

```
#include <stdio.h>
#define READ 1
#define WRITE 2
#define EXECUTE 4
#define VALID_PERMS (READ | WRITE | EXECUTE)
int get_perm(int user_perm, int perm_type){
    return user_perm ① perm_type;
}
void grant_perm(int *user_perm, int perm_type){
    if ((perm_type & VALID_PERMS) != perm_type){
        printf("Not valid permission %n"); return;
    }
    *user_perm= *user_perm ② perm_type;
}
void revoke_perm(int *user_perm, int perm_type){
    *user_perm= *user_perm ③ perm_type;
}
void show_perm(int user_perm, int perm_type){
    const char *perm_text;
    switch (perm_type){
        case READ: perm_text="read"; break;
        case WRITE: perm_text="write"; break;
        case EXECUTE: perm_text="execute"; break;
        default: printf("Unknown permission %n"); return;
    }
    if(get_perm(user_perm,perm_type) == 0){
        printf("No %s permission %n",perm_text);
    }
    else{
        printf("Has %s permission %n",perm_text);
    }
}
int main(void) {
    int user_perm= EXECUTE | READ;
    show_perm(user_perm,WRITE); //No write permission と出力される
    revoke_perm(&user_perm,EXECUTE);
    grant_perm(&user_perm,WRITE);
    show_perm(user_perm,EXECUTE); //No execute permission と出力される
    show_perm(user_perm,WRITE); //Has write permission と出力される
    return 0;
}
```

(2) 次に示す関数 func1 は、スネークケースで記述された文字列(例: get_num)を小文字始まりのキャメルケース(例: getNum)に変換する関数である。変換規則は、_の直後の小文字を大文字に変換し、_を削除するものとする。_の直後が大文字または非アルファベット文字であれば変換しない。空欄 ①～③を埋めて、プログラムを完成させなさい。

```
#include <stdio.h>
void func1(char *s) {
    char *q=s;
    for (char *p = s; *p!= '\0'; p++) {
        if(*p=='_'){
            if ( ① >= 'a' && ① <= 'z') {
                *q= ② ;
            }
            else{
                *q=*(p+1);
            }
            p++;
        }
        else {
            ③ ;
        }
        q++;
    }
    *q= '\0';
}

int main(void) {
    char variable_name[] = "first_and_last_name";
    func1(variable_name);
    printf("%s \n", variable_name); //firstAndLastName と出力される
    return 0;
}
```

(3)次に示す関数 `rgb_to_gray` は、高さ `HEIGHT`、幅 `WIDTH` の RGB 画像データ (3 次元配列) をグレースケール画像データ (2 次元配列) に変換する関数である。RGB の各要素は 0~255 の整数で格納されており、グレースケール値は $(R + G + B) / 3$ で計算する。空欄 ①~⑤ を埋めて、プログラムを完成させなさい。

```
#include <stdio.h>

#define HEIGHT 2
#define WIDTH 2
#define COLORS 3

void rgb_to_gray(int (*src)[WIDTH][COLORS], int (*dst)[WIDTH]){
    for (int i = 0; i < HEIGHT; i++) {
        for (int j = 0; j < WIDTH; j++) {
            int sum = 0;
            for (int k = 0; k < ① ; k++) {
                sum += ② ;
            }
            ③ = ④ ;
        }
    }
}

int main(void) {
    int rgb[HEIGHT][WIDTH][COLORS] = {
        {{110, 80, 20}, {100, 50, 6}},
        {{0, 249, 0}, {0, 0, 210}}
    };
    int gray[HEIGHT][WIDTH];
    rgb_to_gray(rgb, ⑤ );
    for (int i = 0; i < HEIGHT; i++) {
        for (int j = 0; j < WIDTH; j++) {
            printf("%d ", gray[i][j]);
        }
        printf("\n");
    }
    //70 52
    //83 70 と出力される
    return 0;
}
```

(4) 関数 func2 は、重複する要素がない昇順ソート済み連結リスト（集合を表す）を 2 つ受け取って処理を行う。次に示す C 言語プログラムについて、問（ア）（イ）に答えなさい。

```
#include <stdio.h>
struct Element{
    int num;
    struct Element *next;
};

void func2(struct Element *a, struct Element *b) {
    if (a == NULL || b == NULL) {
        return;
    }

    if (a->num == b->num) {
        printf("%d ", a->num);
        func2(a->next, b->next);
    }
    else if (a->num < b->num) {
        func2(a->next, b);
    }
    else{
        func2(a, b->next);
    }
}

int main(void) {
    struct Element a_num1 = {2, NULL}, a_num2 = {4, NULL},
    a_num3 = {5, NULL}, a_num4={7, NULL};
    a_num1.next = &a_num2;
    a_num2.next = &a_num3;
    a_num3.next = &a_num4;

    struct Element b_num1 = {1, NULL}, b_num2 = {3, NULL},
    b_num3 = {4, NULL}, b_num4={5, NULL};
    b_num1.next = &b_num2;
    b_num2.next = &b_num3;
    b_num3.next = &b_num4;

    func2(&a_num1, &b_num1);
    return 0;
}
```

(ア) このプログラムを実行したとき、出力される内容を示しなさい。

(イ) 関数 func2 はどのような機能をもつか 30 文字以内で説明しなさい。

Ⅱ

(1) 以下の文章の空欄を適切な 10 進数値で埋めなさい。

- 主記憶装置の容量が 512 メガバイトであり、1 バイトごとに固有のアドレスを割り当てるバイトアドレッシング方式を採用しているコンピュータを考える。この主記憶装置内のすべてのアドレスを表現するために必要なアドレス長は (a) ビットである。
- 標本化周波数 48KHz、量子化ビット数 24 ビット、ステレオ 2 チャンネルで PCM 録音した音声データを考える。圧縮を行わないものとする、1 分間の録音データ容量は (b) バイトになる。
- 1 文字を 3 バイト固定長で表現する文字コード体系を考える。ただし、3 バイトのうち上位 4 ビットは将来の拡張用として予約されており、文字の識別には使用しないものとする。この文字コード体系で表現可能な文字数の最大値は (c) 文字である。
- 整数部・小数部を含む 2 進固定小数点数 1101.1101 を 10 進数に変換すると (d) である。

(2) 以下の文章の空欄を適切な 2 進数値で埋めなさい。

- 1 の補数表現を用いて、10 進数の -0 を 8 ビット固定長の 2 進数で表現すると、(a) となる。
- 2 の補数表現を用いて、10 進数の -76 を 8 ビット固定長の 2 進数で表現すると、(b) となる。この値を 3 ビット算術右シフトした結果と 2 ビット循環左シフトした結果はそれぞれ (c) と (d) となる。

(3) コンピュータのメモリ装置について、以下の文章の空欄を適切な術語で埋めなさい。

- プロセッサ内部にある (a) は、演算対象となるデータや演算結果を一時的に保持する記憶装置である。
- 一方、実行中のプログラムの命令列やデータは (b) に格納されている。
- 通常、(a) と (b) の間には大きな速度差が存在するため、両者の間に (c) を配置する。
- (c) には、頻繁に使用される (d) や (e) が一時的に格納される。
- (c) はプログラムのアクセスにおけるメモリ参照の (f) 性を利用することで、メモリアクセス性能を向上させている。また、メモリ参照の (f) 性は、ファイル装置を利用して (b) の容量を見かけ上拡大する (g) という技術でも活用されている。

(4) サーバ室の強制冷却装置を制御する論理回路を考える。入力 a , b , c , d と出力 Y を以下のように定義する。

- 入力 a : サーバ室が高温状態であるときに論理値 1、高温状態でなければ論理値 0
- 入力 b : サーバ群が高負荷状態であるときに論理値 1、高負荷状態でなければ論理値 0
- 入力 c : サーバ室内で管理者が作業中であるときに論理値 1、管理者が作業中でなければ論理値 0
- 入力 d : サーバ室の消火設備が作動中であるときに論理値 1、作動中でなければ論理値 0
- 出力 Y : 強制冷却装置を動作させるときに論理値 1、停止させるときに論理値 0

サーバ室の強制冷却装置は以下の条件に従って動作する。

1. サーバ群が高負荷状態であるときにサーバ室が高温状態である
2. または、サーバ室が高温状態であり、サーバ室内で管理者が作業中である
3. ただし、サーバ室の消火設備が作動中である場合は動作しない

以下の問いに答えなさい。ただし、使用する論理演算子として AND 演算子「 $\cdot$ 」、OR 演算子「 $+$ 」、NOT 演算子「 $\bar{n}$ 」(n は論理変数または論理式)とする。論理演算子の優先順位は高いものから NOT、AND、OR の順とする。

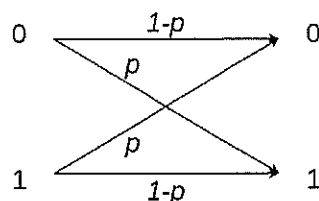
- (a) 論理関数 $Y(a, b, c, d)$ を表す標準積和形の論理式を示しなさい。
- (b) 論理関数 $Y(a, b, c, d)$ に対する最小積和形の論理式を示しなさい。
- (c) 論理関数 $Y(a, b, c, d)$ に対する最小和積形を NOR 演算子「 $\downarrow$ 」のみを用いた論理式で示しなさい。ここで、 $\bar{n} = n \downarrow n$ とする。

III

- (1) 以下に示す無記憶情報源 U について以下の間に答えなさい。ただし、 $\log_{10} 2 = 0.30$ 、 $\log_{10} 3 = 0.48$ としなさい。

メッセージ記号	メッセージ内容	生起確率
0	当選	0.25
1	落選	0.75

- (ア) 情報源 U の平均情報量 $H(U)$ [ビット] を求めなさい。
- (イ) 情報源 U を以下に示す 2 元対称通信路を用いて伝送するものとする。 $p = 0.25$ のとき、この通信路を通じて受信するメッセージを V とするとき、情報源 U に対する条件付平均情報量 $H(U|V)$ [ビット] を求めなさい。



- (ウ) 上記の 2 元対称通信路 ($p = 0.25$) の通信路容量 C [ビット/記号] を求めなさい。
- (2) 通信路の符号化に関する以下の文章の主張は正しいか、正しくないか、答えなさい。正しくない場合には、どのように修正すればよいか、答えなさい。
- 『通信路容量が C [ビット/記号] の離散的無記憶通信路を用いて通信を行うとき、適当に情報を符号化することにより、通信の信頼度をいくらでも高くすることができるが、そのかわりに情報伝送速度 R [ビット/記号] は限りなく小さくなる。』
- (3) 公開鍵暗号を用いて通信を行うとき、通信相手から送られる公開鍵の信頼性（その公開鍵が想定している通信相手の正しい公開鍵かどうか）を担保するためにどのような方法が用いられているか、説明しなさい。