

|      |      |
|------|------|
| 検査科目 | 情報基礎 |
|------|------|

問題 1

以下の各問いに答えよ。

- (1) 10進数  $-100$  を8ビットの2進数に変換し答えよ。ただし、負数の表現方法としては2の補数表現を使用すること。
- (2) 8進の小数  $6.74$  を10進の小数に変換し答えよ。
- (3) 説明文「無限級数で表される数値を計算する際に、最初の有限個の項だけで計算を止めることにより生じる誤差」が説明している最も適切な用語を、以下の選択肢ア～エから1つ選び記号で答えよ。

ア. 丸め誤差    イ. 打ち切り誤差    ウ. 情報落ち    エ. 桁落ち

- (4) 集合  $A, B, C$  に対し、集合  $(A - C) \cup (B - C)$  を (  ) の部分で表しているベン図を、図1の選択肢ア～エから1つ選び記号で答えよ。ただし、 $X, Y, Z$  を集合とすると、 $X \cap Y$  は  $X$  と  $Y$  の積集合（共通部分）、 $X \cup Y$  は  $X$  と  $Y$  の和集合、 $X^c$  は  $X$  の補集合をそれぞれ表し、集合  $X - Y$  は  $X \cap Y^c$  で定義される集合とする。

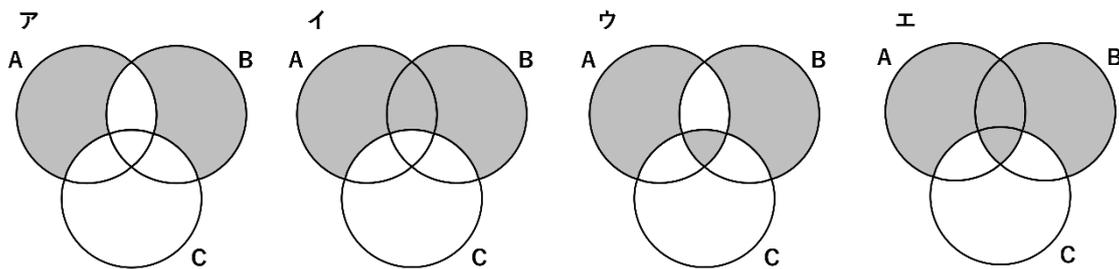


図 1. ベン図

- (5) 関数  $f(n)$  は、自然数  $0, 1, 2, 3, \dots$  上で、以下のように再帰的に定義されているとする。このとき、 $f(10)$  の値を答えよ。

$$f(n) = \begin{cases} 0 & (n = 0) \\ 1 & (n = 1) \\ f(n-1) + f(n-2) & (n \geq 2) \end{cases}$$

|      |      |
|------|------|
| 検査科目 | 情報基礎 |
|------|------|

問題2

図2は3ビットの入力 ( $b_2, b_1, b_0$ ) に対し、隣り合う2つ以上のビットの値が「1」である場合は「1」を出力( $F$ )し、そうでない場合は「0」を出力( $F$ )する回路（連続する「1」の検出回路）である。このとき、以下の間に答えよ。

- (1) 表1は連続する「1」の検出回路の真理値表である。空欄(a)～(d)に入る値をそれぞれ答えよ。
- (2) 連続する「1」の検出回路の出力( $F$ )を与える論理式を最も簡略化した積和形で答えよ。

表 1. 連続する「1」の検出回路の真理値表

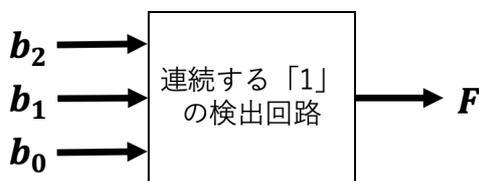


図 2. 連続する「1」の検出回路

| $b_2$ | $b_1$ | $b_0$ | $F$ |
|-------|-------|-------|-----|
| 0     | 0     | 0     | 0   |
| 0     | 0     | 1     | 0   |
| 0     | 1     | 0     | 0   |
| 0     | 1     | 1     | (a) |
| 1     | 0     | 0     | (b) |
| 1     | 0     | 1     | 0   |
| 1     | 1     | 0     | (c) |
| 1     | 1     | 1     | (d) |

問題3

リスト1のプログラムは、入力されたデータから二分木を生成するものである。節内の値は0以上の整数とし、二分木の全ての部分木で図3が成立するものとしたとき、次の問いに答えよ。

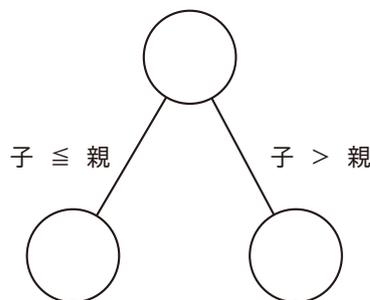
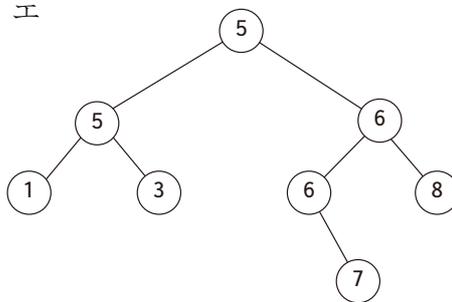
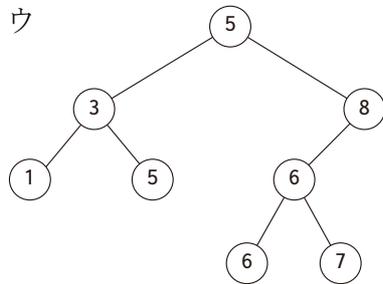
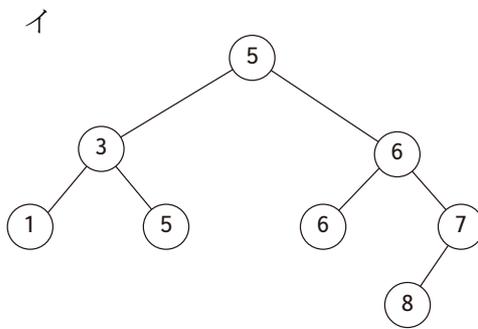
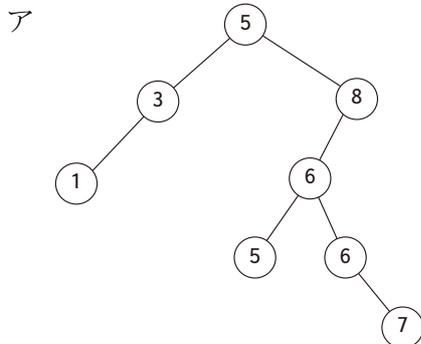


図 3. 二分木の生成条件

- (1) リスト1では、二分木の節を構成する型として `node` を定義している。このような、異なる型の変数をひとまとめにしたものの名称を答えよ。
- (2) リスト1中の空欄 (A)から(D)に入る適切なコードを答えよ。

|      |      |
|------|------|
| 検査科目 | 情報基礎 |
|------|------|

(3) プログラム実行時に「5 3 8 6 5 6 1 7」と入力した時、生成される二分木の形は次のどれになるか、以下の選択肢ア～エから1つ選び記号で答えよ。



(4) 木の高さが $h$ である完全二分木の節数はいくつになるか。 $h$ を用いた式で表せ。ただし、総和を表す記号（ $\Sigma$ ）を使用せず、逐次的な加算の形とすること。

「木の高さ」とは、根から葉までにたどる辺の数であり、図3の高さは1である。

(5) 二分木で利用したメモリ領域解放を再帰で実現する関数 `void free_tree(node *)` の処理で、正しいものを次のア～ウのコードから1つ選択し、記号で答えよ。

|   |   |   |
|---|---|---|
| <p>ア</p> <pre> if(p != NULL) {     free_tree(p-&gt;left);     free_tree(p-&gt;right);     free(p); }                 </pre> | <p>イ</p> <pre> if(p != NULL) {     free_tree(p-&gt;left);     free(p);     free_tree(p-&gt;right); }                 </pre> | <p>ウ</p> <pre> if(p != NULL) {     free(p);     free_tree(p-&gt;left);     free_tree(p-&gt;right); }                 </pre> |
|---|---|---|

|      |      |
|------|------|
| 検査科目 | 情報基礎 |
|------|------|

リスト 1. 二分木生成プログラムのソースコード

```

#include<stdio.h>
#include<stdlib.h>

typedef struct cell {
    int value;
    (A) *left;
    (A) *right;
} node;

void insert_tree(node **p, int data);
void free_tree(node *p);

int main(void) {
    node *root = NULL;
    int data;

    while(1) {
        scanf("%d", (B));

        if(data < 0) break;
        insert_tree((C), data);
    }

    free_tree(root);

    return 0;
}

void insert_tree(node **p, int data) {
    if(*p == NULL) {
        *p = (node *)malloc(sizeof(node));
        (D) value = data;
        (D) left = NULL;
        (D) right = NULL;
    } else if(data <= (D) value) {
        insert_tree(&(D) left, data);
    } else {
        insert_tree(&(D) right, data);
    }
}

```

|      |      |
|------|------|
| 検査科目 | 情報基礎 |
|------|------|

```
void free_tree(node *p) {  
    (5)  
}
```