

問 題 訂 正

10 ページ 専門科目 B 群 プログラミング

問題 1 (2) [選択肢] イ

(誤) N U L L

(正) p

令和 8 年度 専攻科入学者選抜学力検査問題

100点
90分

専門科目 (電子情報通信工学専攻)

A 群 電磁気学・電気回路・電子回路

B 群 デジタル回路・情報基礎・プログラミング

注意事項

1. 問題用紙は指示があるまで開かないこと。
2. 問題用紙は 1 ページから 11 ページまでである。
検査開始の合図のあとで確かめること。
3. 解答は、すべて解答用紙に記入すること。
4. 解答用紙の得点欄には記入しないこと。
5. A 群または B 群のいずれかを選択し、選択した群の
すべての科目を解答すること。
6. 配点は A 群または B 群の 300 点を 100 点に換算する。

専門科目 A 群

電磁気学

問題 1

半径 a [m] の円環に正 (+) の電荷 Q [C] が一様に分布している。円環の中心軸上 x [m] の点を点 P とする（図 1.1 参照）とき、以下の問いに答えなさい。なお、円環は真空中に置かれている。真空の誘電率は ϵ_0 [F/m] とすること。

- (1) 点 P の電位 V_P [V] を求めなさい。
- (2) 点 P における中心軸方向の電界 E_x [V/m] を求めなさい。

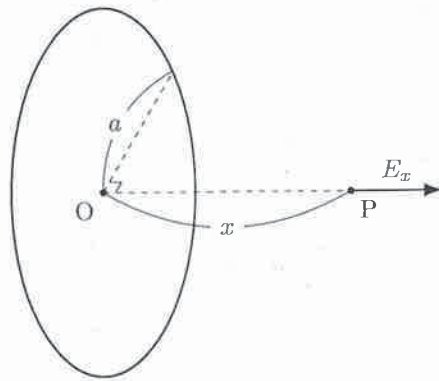


図 1.1：一様な電荷分布を持つ円環

問題 2

面積 S [m²] の薄い平板導体 A, B が真空中に d [m] 間隔で平行に置かれている。 d は導体の縦横の長さに比べ十分に小さいとする。いま、導体 A に正 (+) の電荷 $+Q$ [C] を与え、導体 B に負 (-) の電荷 $-Q$ [C] を与えたとき、以下の問いに答えなさい。真空の誘電率は ϵ_0 [F/m] とすること。

- (1) 導体間に生じる電界 E [V/m] を求めなさい。
- (2) 導体間の電位差 V_{AB} [V] を求めなさい。
- (3) 導体間の静電容量 C [F] を求めなさい。
- (4) 導体間に蓄えられた静電エネルギー W [J] を求めなさい。

電磁気学

問題 3

内半径 a [m]，外半径 b [m] の無限長中空導体が真空中に置かれている。図 3.1 はその断面を表している。この導体に図 3.1 のように電流 I [A] が一様に流れている。中心からの距離 r [m] の位置における，磁束密度を B [T] とするとき，以下の問いに答えなさい。真空の透磁率は μ_0 [H/m] とすること。

- (1) 中空部分 ($0 \leq r \leq a$) の磁束密度 B [T] を求めなさい。
- (2) 導体部分 ($a < r \leq b$) の磁束密度 B [T] を求めなさい。
- (3) 導体外部 ($b < r$) の磁束密度 B [T] を求めなさい。

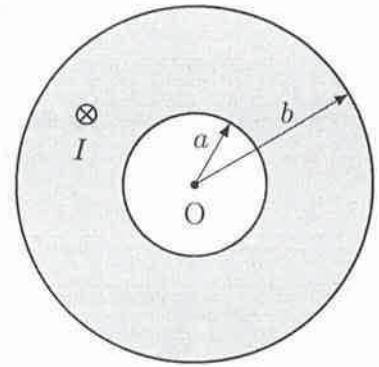


図 3.1： 無限長中空導体の断面

専門科目 A 群

電気回路

問題 1

図 1.1 に示す回路において正弦波交流電圧源の複素電圧を $E = 100 \text{ V}$ 、角周波数を $\omega = 200 \text{ rad/s}$ とする。また、 $R = 30 \Omega$ 、 $L = 200 \text{ mH}$ とする。以下の問いに答えよ。

- (1) 回路に流れる複素電流 I を求めよ。
- (2) 複素電圧 V_R を求めよ。
- (3) 複素電圧 V_L を求めよ。
- (4) 回路の有効電力 P を求めよ。

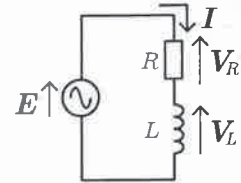


図 1.1: 問題 1 の回路

問題 2

図 2.1 に示す回路において正弦波交流電圧源の複素電圧を $E = 100 \text{ V}$ とする。また、 $R = 10 \Omega$ 、 $C = 50 \mu\text{F}$ とする。以下の問いに答えよ。

- (1) 電圧源の角周波数を $\omega = 2000 \text{ rad/s}$ とするとき、回路の合成インピーダンス Z を求めよ。
- (2) 電圧源の角周波数を $\omega = 2000 \text{ rad/s}$ とするとき、回路に流れる複素電流 I を求めよ。
- (3) 角周波数を 0 rad/s から無限大まで変化させると複素平面における Z のベクトルの先端は図 2.2 の太線で示す半円の円周上を通る。これを Z のベクトル軌跡と呼ぶ。この円周上にあり虚部の値が最小となる点 A の実部と虚部の値を求めよ。

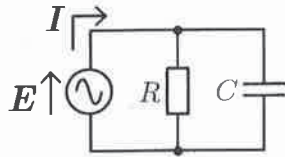


図 2.1: 問題 2 の回路

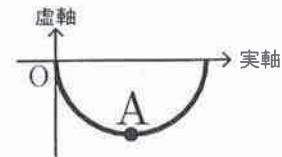


図 2.2: 問題 2 の Z のベクトル軌跡

問題 3

図 3.1 に示す回路において正弦波交流電圧源の複素電圧を $E = 100 \text{ V}$ とする。また、 $R = 10 \text{ k}\Omega$ 、 $L = 250 \text{ mH}$ 、 $C = 100 \mu\text{F}$ とする。以下の問いに答えよ。

- (1) 回路の共振角周波数 ω_0 を求めよ。
- (2) 共振時の複素電流 I を求めよ。
- (3) 回路の Q 値（尖鋭度）を求めよ。

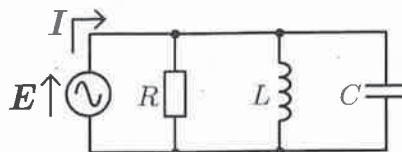


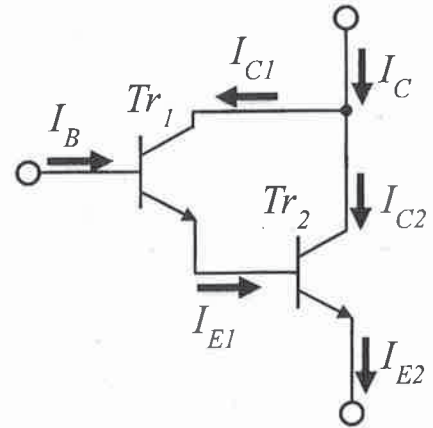
図 3.1: 問題 3 の回路

専門科目 A 群

電子回路

問題 1

図 1.1 のトランジスタ回路について以下の問いに答えなさい。ただし、トランジスタ Tr_1 と Tr_2 の電流増幅率はそれぞれ h_{FE1} , h_{FE2} である。また、トランジスタ Tr_1 と Tr_2 のバイアスは能動領域に設定されているとする。



- (1) Tr_1 について、ベース電流 I_B が入力されたとき、コレクタ電流 I_{C1} およびエミッタ電流 I_{E1} はそれぞれどのようなになるか、 I_B と h_{FE1} を用いた式で示しなさい。
- (2) Tr_2 のコレクタ電流 I_{C2} およびエミッタ電流 I_{E2} はそれぞれどのようなになるか、 I_B と h_{FE1} および h_{FE2} を用いた式で示しなさい。
- (3) 図中の電流 I_C はどのようなになるか、 I_B と h_{FE1} および h_{FE2} を用いた式で示しなさい。
- (4) 回路全体の電流増幅率 $h_{FE} = I_C / I_B$ はどのようなになるか、 h_{FE1} および h_{FE2} を用いた式で示しなさい。

図 1.1：トランジスタ回路

問題 2

図 2.1 のトランジスタのバイアス回路について以下の問いに答えなさい。ただし、トランジスタ Tr の能動領域でのベース-エミッタ間電圧 $V_{BE} = 0.7 \text{ V}$ 、コレクタ-エミッタ間電圧 $V_{CE} = 5.2 \text{ V}$ 、電流増幅率 $h_{FE} = 160$ 、コレクタ電流 $I_C = 6.0 \text{ mA}$ 、抵抗 $R_1 = 200 \text{ k}\Omega$ 、抵抗 $R_2 = 200 \text{ }\Omega$ 、直流電源電圧 $E = 10 \text{ V}$ とする。

- (1) ベース電流 I_B を求めなさい。
- (2) 抵抗 R_3 の値を求めなさい。
- (3) 抵抗 R_4 の値を求めなさい。

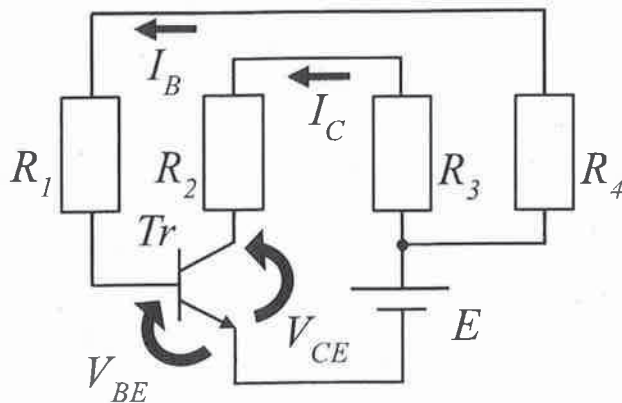


図 2.1：バイアス回路

専門科目 A 群

電子回路

問題 3

図 3.1 の回路について以下の問いに答えなさい。ただし、オペアンプは理想特性とする。

- (1) 節点 a-b 間の電圧 V_S はいくらになるか答えなさい。
- (2) 入力インピーダンス Z_{in} はいくらになるか答えなさい。
- (3) 入力電流 I_{in} と出力電圧 V_{out} の関係式を求めなさい。

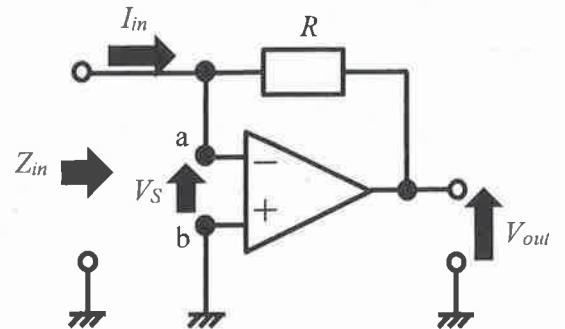


図 3.1：オペアンプ回路

問題 4

図 4.1 および図 4.2 の回路について以下の問いに答えなさい。ただし、図中の入力電圧 $e(t)$ は周波数 f [Hz]、実効値 E [V] の正弦波電圧である。また、図中のダイオードは全て理想特性とする。

- (1) 図 4.1 において、抵抗 R の両端電圧 V_R の最大値、最小値、平均値を求めなさい。
- (2) 図 4.2 は、図 4.1 の抵抗 R をコンデンサ C へ変更した回路である。十分に時間が経過した後、コンデンサ C の両端電圧 V_C はいくらになるか求めなさい。

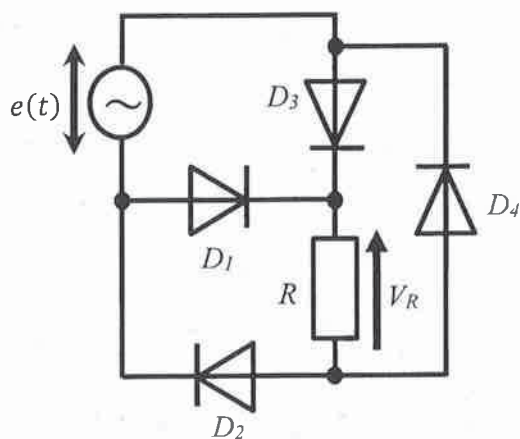


図 4.1：ダイオードと抵抗の回路

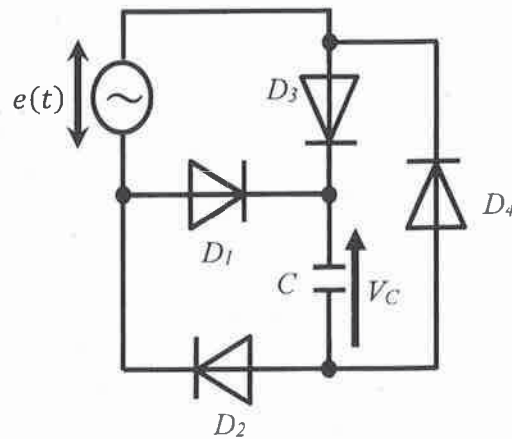


図 4.2：ダイオードとコンデンサの回路

専門科目 B 群

デジタル回路

問題 1

以下の問 (1)～(3) に答えよ。

- (1) 変数 n は、 $-64 \leq n \leq 64$ の範囲の値をとる。符号付き 2 進数を用いて n を表すためには、最低何ビット必要であるか答えよ。
- (2) 2 進数 10110100 が 8 ビットの 2 の補数を用いて表現された符号付き整数であるとき、その絶対値を 10 進数で答えよ。
- (3) 10 進数で表された計算式 $10 - 5$ を、8 ビットの 2 進数で表し、補数加算により計算せよ。計算過程を示し、答は 2 進数で表示せよ。

問題 2

論理関数 $Z = (\bar{A} + B + \bar{C}) \cdot (\bar{A} + \bar{B} + C) \cdot (B + C)$ について、以下の問 (1)～(3) に答えよ。

- (1) 論理関数 Z の真理値表を書け。
- (2) Z を簡単化して加法形を求めよ。
- (3) (2)で得られた論理関数を、図 2.1 に示す基本論理素子（AND, OR, NOT）のみを用いた回路として図示せよ。

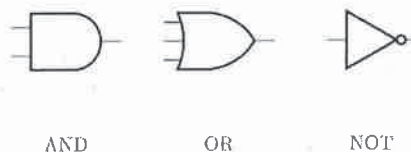


図 2.1：基本論理素子の記号

専門科目 B 群

情報基礎

問題 1

パイプライン処理に関する以下の問いに答えよ。

- (1) 1 命令を 5 ステージで実行する理想的なパイプライン処理において 4 命令を実行したときパイプライン処理のない実行と比べて何倍の高速化が可能となるか答えよ。
- (2) パイプライン処理におけるハザードにはどのようなものがあるか答えよ。

問題 2

表 2.1 は、ある CPU の命令セット中の命令を実行に必要なクロック数によって分類し、それぞれに属する命令が命令セット全体に占める割合を示したものである。この CPU が動作クロック周波数 2 GHz で動作するとき、以下の問いに答えよ。

表 2.1：実行に必要なクロック数と全命令に対する割合

命令の分類	実行に必要なクロック数	割合(%)
A	4	60
B	6	30
C	8	10

- (1) A に属する命令を 1 命令行うために必要な時間を答えよ。
- (2) 1 命令を実行するのに必要な平均クロック数を答えよ。
- (3) 1 秒間に実行可能な命令数の平均を答えよ。

問題 3

すべての命令はキャッシュメモリから読み込まれると仮定して、キャッシュヒット率 90 %、キャッシュメモリのアクセス時間 20 ns、主記憶装置とキャッシュメモリ間のアクセス時間 100 ns であるとき実効アクセス時間を求めよ。

問題 4

情報セキュリティについて、以下の問いに答えよ。

- (1) 機密性とは何か答えよ。
- (2) 完全性とは何か答えよ。
- (3) 可用性とは何か答えよ。
- (4) 情報技術などを用いずに、人間の心理や盲点について、パスワードなどの機密情報を入手する方法のことを何と呼ぶか。次の (A) から (D) の中から選び、記号で答えよ。
 (A) フィッシング (B) マルウェア感染 (C) ソーシャルエンジニアリング
 (D) DDoS 攻撃

プログラミング

問題 1

あるシステムでは、動的に追加される整数データを効率よく管理するために二分探索木 (Binary Search Tree : BST) を用いている。

この BST では、以下のルールに従ってノードが配置される。

- 任意のノード x は整数値を 1 つ保持する。
- ノード x の左部分木には自ノードよりも小さいノードが接続される。
- ノード x の右部分木には自ノードよりも大きいノードが接続される。

C 言語を用いて、この二分探索木 (BST) のプログラムを作成する。木構造を構成するノードの構造体は図 1.1 のように定義されている。グローバル変数 `root` はルートノードを指すポインタであり、ノードが 1 つもない状態として `NULL` に初期化する。

```
typedef struct node {  
    int data;           // ノードが保持する整数データ  
    struct node *left;  // 左の子ノードへのポインタ  
    struct node *right; // 右の子ノードへのポインタ  
} NODE;  
  
NODE *root = NULL;      // ルートノードへのポインタ（初期は NULL）
```

図 1.1 : ノードの構造体およびルートノードへのポインタの定義

このプログラムに、以下の「探索」「挿入」「削除」のそれぞれの処理を行う関数を定義する。

【探索処理】

関数 `search` は、BST から引数で与えられた整数 `key` を持つノードを探し、そのノードへのポインタを返す関数である。該当ノードが存在しない場合は `NULL` を返す。

探索は以下を行う：

1. 注目するノードを指すポインタを p とし、 p がルートノードを指すように初期化する。
 p が指すノードの値と `key` を比較しながら進む。
2. 注目するノードの値と `key` が一致した場合は、そのノードへのポインタを返す。
3. 注目するノードの値より `key` が小さい場合は左へ、大きい場合は右へ p を進める。
4. 最後まで見つからなかった場合は探索失敗として `NULL` を返す。

【挿入処理】

関数 `insert` は、引数 `key` に指定された整数値を持つノードを BST のルールに従って適切な位置に挿入する。この関数は返り値として挿入したノードへのポインタを返す。ただし、既に同じ値が存在する場合には、ノードの挿入を行わず `NULL` を返す。

専門科目 B 群

プログラミング

この関数 insert を用いて図 1.2 に示す呼び出しを行ったときの BST を図 1.3(a) に示す。この状態に更に値 15 を挿入した場合には、図 1.3(b) のように遷移する。

```
insert(10);
insert(6);
insert(12);
insert(7);
insert(2);
```

図 1.2 : insert の例

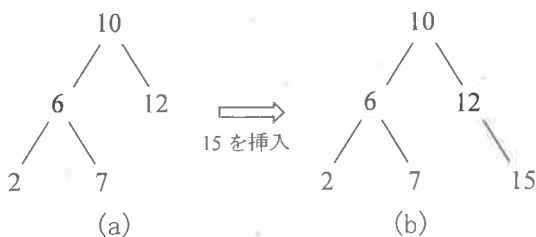


図 1.3 : 生成される木構造の簡略図

【削除処理】

関数 delete は、引数 key で指定された整数値を持つノードを BST から削除する。削除対象のノードが存在しない場合は何もせず 0 を返し、削除に成功した場合は 1 を返す。

削除の対象ノードに応じて、以下の 3 つのケースが考慮される：

1. 子ノードがない場合： 単にノードを削除する。
2. 子ノードを 1 つだけ持つ場合： その子を親に直接つなげて削除する。
3. 子ノードを 2 つ持つ場合： 右部分木に含まれる最小のノードで置き換える。

図 1.4 に各々のケースの削除例を示す。

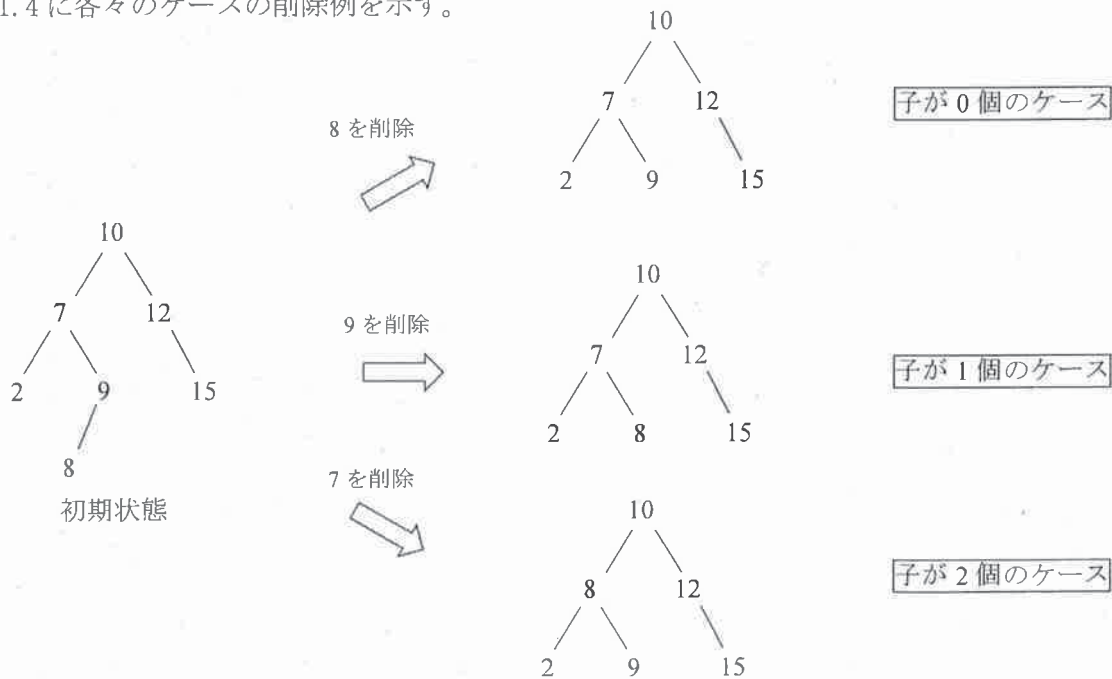


図 1.4 : 子ノードの数の違いによる削除例

専門科目 B 群

プログラミング

- (1) ノードが 1 つもない状態から、右に示す関数呼び出しの順番に BST に操作を加えたとき、最終的な木構造はどうか。この操作をすべて行った後の木構造を簡略図で示せ。

```
insert(50);
insert(30);
insert(70);
insert(20);
insert(40);
delete(30);
insert(60);
insert(30);
delete(40);
```

- (2) 次のプログラムは関数 search の定義部である。

使用されている関数 cmpdata(a, b) は、2 つの整数値 a, b を引数にとり、 $a=b$ のとき 0 を、 $a < b$ のときに負値を、 $a > b$ のときは正値を返す。

空欄①～⑤に当てはまる適切な字句を選択肢ア～コから選んで答えよ。

```
1: NODE *search(int key)
2: {
3:     NODE *p = ①;
4:
5:     while (p != NULL) {
6:         int ret = cmpdata(key, ②);
7:         if (ret == 0)
8:             return ③;
9:         else if (ret < 0)
10:            p = ④;
11:         else
12:            p = ⑤;
13:     }
14:     return NULL;
15: }
```

[選択肢]

ア: p++	イ: NULL	ウ: &root	エ: p->data	オ: p->left
カ: (*p)->left	キ: p->right	ク: (*p)->right	ケ: root	コ: p--

- (3) 次のプログラムは、BST にノードを追加する関数 insert の定義部である。この関数では、動的に確保した新しいノードを適切な位置に挿入する。挿入するには、親ノードのポインタ (left や right メンバまたは root) が挿入ノードを指すように書き換える必要があるため、ポインタのポインタ p を使って、親ノードのポインタを間接的に操作している。p の更新の条件判断には(2) で用いた関数 cmpdata(a, b) を用いる。

空欄①～⑤に当てはまる適切な字句を答えよ。

プログラミング

```

1: NODE *insert(int key)
2: {
3:     NODE **p = ①;
4:
5:     while (*p != NULL) {
6:         int ret = cmpdata(key, (*p)->data);
7:         if (ret == 0)
8:             return ②;
9:         else if (ret < 0)
10:            p = ③;
11:         else
12:            p = ④;
13:     }
14:
15:     NODE *new = malloc(sizeof(NODE));
16:     new->data = key;
17:     new->left = NULL;
18:     new->right = NULL;
19:     ⑤ = new;
20:
21:     return new;
22: }
    
```

(4) 次に示す3つの関数は、BTSの全てのノードに順に立ち寄る訪問関数である。

図1.5の木構造に対して、それぞれの関数にルートノードへのポインタを引数に与えたときの出力結果を答えよ。

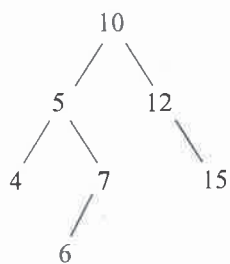


図1.5：対象の木構造

```

1: void preorder(NODE *p)
2: {
3:     if (p == NULL) return;
4:
5:     printf("%d ", p->data);
6:     preorder(p->left);
7:     preorder(p->right);
8: }
    
```

プログラムA

```

1: void inorder(NODE *p)
2: {
3:     if (p == NULL) return;
4:
5:     inorder(p->left);
6:     printf("%d ", p->data);
7:     inorder(p->right);
8: }
    
```

プログラムB

```

1: void postorder(NODE *p)
2: {
3:     if (p == NULL) return;
4:
5:     postorder(p->left);
6:     postorder(p->right);
7:     printf("%d ", p->data);
8: }
    
```

プログラムC