受験番号＿＿＿＿＿＿＿＿＿＿ 氏名 ＿＿＿＿＿＿＿＿＿＿＿＿ 評点：＿＿＿＿＿

1 右のように単方向リストのためのクラスが与えられているとする．次の問題に答えよ．

（問題 1,2 のコードは Python 風の疑似コードである．コードのブロック構造はインデントにより表されている．"def" は関数の定義であり，"self"はインスタンス自身への参照である．"__init__" はコンストラクタであり，インスタンス（具体的なデータの入れ物）生成時に実行される．インスタンス生成は new_node = Node(data) のようにして行う．各行の'#'以下はコメントである．）

```
class Node:
    def __init__(self, data): # 新しいノードを作るときに呼ばれる
        self.data = data # ノードに値を入れる
        self.next = None # 次のノードへのつながり（最初はない）

class SinglyLinkedList:
    def __init__(self): # 新しいリストを作るときに呼ばれる
        self.head = None # 先頭ノード（最初は無い）
        self.size = 0

    def append(self, data):          # 末尾に新しいノードを追加する
        new_node = Node(data)        # 新しいノードを作る
        if _____(a)_____ :           # リストが空の場合
            self.head = ____(b)____
        else:                        # リストにすでに要素がある場合
            current = self.head
            while current.next is not None:
                current = current.next
            current.next = ____(c)____
        self.size += ___(d)____
```

ここで

・**Node** は、要素値を保持する data と，次のノードを指す参照 next をもつ．

・**SinglyLinkedList** は，先頭ノードを指す head 参照と，要素数 size をもつ単方向連結リストである．

・**append** は，与えられた data をもつ新しいノードをリスト**末尾**に追加する操作である．

(1) 空欄 (a)〜(d) に適切な式（擬似コードでも可）を記入せよ．

(2) 空のリストに対して次の操作を順に実行したとき、(i),(ii)の各ステップ直後のリスト lst の状態を図示せよ．各ステップで以下を示すこと：head ポインタの指す先，各ノードの data と next の値，size の値

```
    lst = SinglyLinkedList()
(i) lst.append(20)
(ii)lst.append(10)
```

（解答欄）

(1) (a)＿＿＿＿＿＿＿＿ (b)＿＿＿＿＿＿＿＿ (c)＿＿＿＿＿＿＿＿ (d)＿＿＿＿＿＿＿＿

(2)

(i)

(ii)

受験番号＿＿＿＿＿＿＿＿＿＿ 氏名 ＿＿＿＿＿＿＿＿＿＿＿＿

2 次の疑似コードはリストＡの要素をマージソートによって**昇順**(小さい順)に並べ替えるアルゴリズムを表している.

```
def MergeSort(A):
    if 長さ(A) <= 1:
        return A
    else:
        A1, A2 = 分割(A)    #A を 2 つに分ける
        L1 = (ア)
        L2 = (イ)

        L = マージ(L1, L2)    # (ウ)
        return L
```

(1) 空欄（**ア**），（**イ**）に入る適切な処理を答えよ.

(2) （**ウ**)の行における「マージ」処理は下記の手順で行う.

下記の空欄（あ）～（え）に入る適切な語句を解答欄の選択肢からそれぞれ選択し丸を付けよ.

※ここで使うL1, L2内の要素は既に昇順に並んでいる.

1. L1 と L2 の＿＿（あ）＿＿の要素を比較し，＿＿（い）＿＿方を新しいリスト L の末尾に追加する.
2. 要素がＬに＿＿（う）＿＿側のリストの＿＿（あ）＿＿の要素を削除する.
3. いずれかのリストが空になるまで＿＿（え）＿＿に戻り，繰り返す.
4. 要素が残っているリストの残りの要素を L にすべて追加する.

(解答欄)

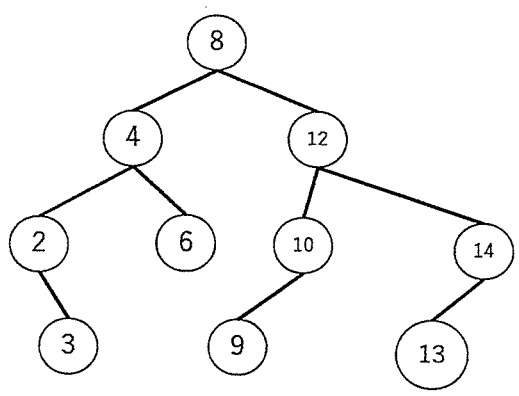
(1) ア　　イ

(2) (あ)　先頭/中央/末尾　（い）　大きい/小さい　（う）　追加された/追加されていない

（え）　1./ 2.

受験番号＿＿＿＿＿＿＿＿＿＿ 氏名 ＿＿＿＿＿＿＿＿＿＿

3 右図の２分木について，以下の問いに答えよ．ただし，いずれの探索においても，同じ深さのノードや子ノードが複数ある場合は左側を先に探索するものとする．



(1) 幅優先探索した場合の探索順序を答えよ．

(2) 後順(帰りがけ順)の深さ優先探索で探索した場合の探索順序を答えよ．

(解答欄)

(1) ＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

(2) ＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

4 データサイズ n に対する異なるアルゴリズムの実行時間が以下の値に比例するとする：

- アルゴリズム A: $5n^2 + 3n + 10$
- アルゴリズム B: $100n + 50$
- アルゴリズム C: $2n \log(n) + n$
- アルゴリズム D: $n^3 + n$

(1) 各アルゴリズムの時間計算量をランダウ記法（オーダー記法，ビッグオー記法とも呼ばれる）で表せ．

(2) $n = 10, n = 100$ の場合のそれぞれについて，どのアルゴリズムが最も効率的（実行時間が最も小さい）か答えよ．
$\log(10) \fallingdotseq 2.3$ を用いてよい．

(3) n が十分に大きい場合，漸近的に最も速いアルゴリズムはどれか答えよ．

(解答欄)

(1) A ＿＿＿＿＿＿ B ＿＿＿＿＿＿ C ＿＿＿＿＿＿ D ＿＿＿＿＿＿

(2) n=10: ＿＿＿＿＿＿ n=100: ＿＿＿＿＿＿

(3) ＿＿＿＿＿＿