



Zin Ko Ko &lt;zinkoko5211@gmail.com&gt;

## Programming DSA Assignment

1 件のメッセージ

Lapyae Phyo &lt;greenarchermoon@gmail.com&gt;

2025年12月3日 11:16

To: zinkoko5211@gmail.com

### 課題

以下のプログラミング言語で、後述のプログラムを作成する。

**Prime** 必ずこれら3種類の言語ではプログラムを作成すること

1. Python
2. JavaScript
3. C

**Optional** 余裕があれば、以下の言語でもプログラムを作成する

4. Perl
5. PHP
6. Ruby
7. Java
8. Bash
9. C++

### 1. Hello World

「hello, world」を出力せよ

### 2. loop

言語の持つ繰り返し(loop)制御構造を用いて

12, 22, 32,, ..., 1002  
の和を求めよ

### 3. 再帰

再帰呼び出しを用いて

12, 22, 32,, ..., 1002  
の和を求めよ

### 4. 構造体

以下の構造体(C)と同等のデータ型を定義し、すべてのメンバーを出力せよ

```
struct st {
    int key;
    char *str;
} st;

st.key=101;
st.str="This is hell.";
```



## 5. リスト

一方向のLinked Listを作成する

1. 構造体で、Linked Listを定義する
2. リスト全体の要素を表示する関数
3. リストをn番目に加える関数 -  $n < 0$  の場合は、末尾の要素
4. n番目の要素を削除する関数 -  $n < 0$  の場合は、末尾の要素
5. 言語仕様としてList型がある場合は、上記に加えて、上記をList型データに対して操作する

## 6. キューとスタック

スタックとキューを作成する

Linked Listを応用して、スタックとキューを実現する

スタック: LIFO(Last In, First Out) のリスト構造

1. push : 値
2. pop :
3. getval :

キュー: FIFO(First In, First Out) のリスト構造

1. enqueue : 値
2. dequeue :
3. getval :

## 7. 木構造

木構造データを作成する

2分木データ構造を作成し、以下のmethodを実装する

```
MyNode {
    int key; // 同一木の中ではUnique
    //char *a; // 要素
    MyNode *left;
    MyNode *right;
}
```



木は、keyが左側から右側に昇順

1. 開始地点以下の木の要素と深さをすべて表示する : 開始地点
2. keyを持つノードを検索する : key
3. 木にkeyを値に持つNodeを1つ加える : key
4. 木からkeyを持つノードを削除する : key

optional. バランスの取れていない木をリバランスする : 木

## 8. ハッシュ関数

キーを受け取り、ハッシュ値を計算する、`sha256`などの既存のハッシュライブラリを用いず、独自のハッシュ関数を作成せよ

## 9. ハッシュ値の衝突回避

ここまでで作成した、ハッシュ関数と木構造を組み合わせて、異なる入力から得られるハッシュ値が衝突した場合に回避する機能を含んだ関数を作成せよ