

2022 年度  
プログラミング第 1 同演習

第 6 回

慶應義塾大学 理工学部 情報工学科

講義担当：河野健二

演習担当：杉浦裕太

## 【課題の提出方法】

提出時期：

- 次の授業日の午前 9:00
- 遅れた場合は減点されます
- 遅れる場合でも、さらにその次の授業日の 9:00 までには提出をしてください（提出フォームが自動的に閉じます）

提出先：

- CANVAS 経由

提出方法：

- 「xxxxx.c」 ファイルを一つずつ CANVAS にアップロードして提出
- 課題を再提出したい場合は、変更した課題だけではなくすべての課題を再アップロードするようにしてください。（以前にアップロードしたものは消えてしまう可能性があります）

ファイル名：

- 課題のスライドのタイトルと同じ（採点時に重要なので必ず同じにしてください）。ただし課題を再アップロードする際はファイル名が変更されることもあり、これに関してはそのままで大丈夫です

終了対応：

- 課題が終わったら各自退出していただいてかまいません
- 授業終了の時間がきて課題が終わっていなくても帰宅していただいてかまいません
- 残りの課題は ITC パソコンか各自のパソコンで行ってください

ITC Linux パソコン以外で作業する際の注意点：

- 自分の環境で発生した問題は自己責任でお願いします
- 提出するプログラムコードは Linux 環境でコンパイルできることを確認すること
- 採点時に、Linux 環境でコンパイルできることを確認するため、その時点でコンパイルが失敗してしまった提出物の点数は自動的に 0 点になってしまいます

## 【必須課題】

### 06-a-01.c

#### 【課題】

整数(int)、次の要素へのポインタ(struct number \*)、前の要素へのポインタ(struct number \*)をメンバとする構造体 number を定義し、双方向リスト構造によって 5 個の整数入力をし、入力順と逆順を出力するプログラムを作成してください。入力数は 5 個と決められてはいるが、配列を用いず、malloc を用いて動的に領域を確保して取り組むこと。

#### 【実行例：入力は赤字】

```
input 5 numbers :  
2  
3  
4  
2  
3  
number list : 2 3 4 2 3  
number list : 3 2 4 3 2
```

#### 【解答例】

## 06-a-02.c

### 【課題】

双方向リストを用いて、EOF を受け取るまで、入力した数値を順番に格納するプログラムを作成せよ。以下の関数を定義して使用せよ。

`void insert(struct data_t *p, struct data_t *w):` p の指す要素の直前に w の指す要素を挿入する。

`void printall(struct data_t *p):` 先頭要素を受けとり、連結されている要素のアドレス、fp, bp のアドレス、データをすべて表示する。

### 【ヒント】

- ・ EOF を受け取るまで整数を受け取るには以下のような for 文を書く

```
int d;
for(;scanf("%d", &d) != EOF ;){
    //受け取った整数に対しての処理//
}
```

- ・ EOF の入力は Ctrl + d （unix 環境）

- ・ printf でアドレスを表示するには%p を使用する。

### 【実行例：入力赤字】

1 2 3 4 5

^d

[p: 0xfca010, bp: 0x7fffe34cc650, fp: 0xfca030, data:1]

[p: 0xfca030, bp: 0xfca010, fp: 0xfca050, data:2]

[p: 0xfca050, bp: 0xfca030, fp: 0xfca070, data:3]

[p: 0xfca070, bp: 0xfca050, fp: 0xfca090, data:4]

[p: 0xfca090, bp: 0xfca070, fp: 0x7fffe34cc650, data:5]

### 【解答例】

## 06-a-03.c

### 【課題】

双方向リストを用いて、入力した数値を昇順に格納するプログラムを作成せよ。以下の関数を定義して使用せよ。

`void insert(struct data_t *p, struct data_t *w)` : `p` の指す要素の前に `w` の指す要素を挿入する。

`struct data_t *insert_pos(struct data_t *p, struct data_t *w)` : 先頭要素 `p` を受け取り、`w` の挿入する位置を返す。

`void printasc(struct data_t *p)`, `void printdsc(struct data_t *p)` : 先頭要素 `p` を受けとり、連結されている要素を昇順、降順で表示する。

### 【実行例：入力は赤字】

```
1 3 4 2 5
^d
12345
54321
```

### 【解答例】

【標準課題】

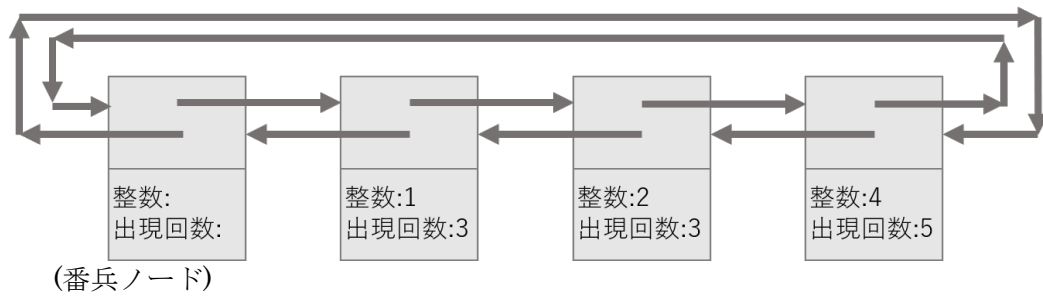
06-b-01.c

【課題】

双方向リストを用いて、EOF を受け取るまで入力した整数とその出現回数を昇順に格納し、出力するプログラムを作成せよ。

【ヒント】

実行例一行目の入力だと図のような双方向リストになるようにせよ。



【実行例：入力は赤字】

```
1 4 4 2 1 2 1 2 4 4 4
^d
[1]: 3
[2]: 3
[4]: 5
---
5 4 4 3 3 3 2 2 2 2 1 1 1 1 1
^d
[1]: 5
[2]: 4
[3]: 3
[4]: 2
[5]: 1
```

【解答例】

## 06-b-02.c

### 【課題】

双方向リストを用いて以下の関数を定義し、キュー構造とスタック構造を両立させるプログラムを作成せよ。なお、配列の使用は禁止する。

`void push(struct data*, int n)`: 番兵ノードを受け取り、値をリストに格納する。

`int pop_queue(struct data*)`: リストをキューとみなして値を取り出す。

`int pop_stack(struct data*)`: リストをスタックとみなして値を取り出す。

`void print_data(struct data*)`: リストに格納されている数値を出力する。

### 【ヒント】

キュー構造とスタック構造について知らない人は各自ウェブで調べてください。

main 関数は以下のように記述すること

```
int main()
{
    int i;
    struct data sentinel;
    sentinel.fp = sentinel.bp = &sentinel;
    push(&sentinel, 1);
    print_data(&sentinel);
    push(&sentinel, 2);
    print_data(&sentinel);
    push(&sentinel, 3);
    print_data(&sentinel);
    pop_queue(&sentinel);
    print_data(&sentinel);
    pop_stack(&sentinel);
    print_data(&sentinel);
    return 0;
}
```

### 【実行例：入力は赤字】

```
push 1
stored nums : 1
push 2
stored nums : 1 2
push 3
stored nums : 1 2 3
pop 1(queue)
stored nums : 2 3
pop 3(stack)
stored nums : 2
```

### 【解答例】

## 06-b-03.c

### 【課題】

課題(06-a-03)に以下の関数

`void remov(struct data_t *p, int index):` 先頭要素(番兵ノード)を受けとり、`index` 番目の要素を消去する(`index` は 0 始まり)。

を追加し、EOF を受け取るまで、入力した数値を順番に双方向リストに格納した後、再び EOF を受け取るまで、入力した数値番目の要素を消去するプログラムを作成せよ。

また、指定した数値番目の要素がない場合はエラーメッセージを表示しなさい。

### 【実行例：入力は赤字】

State Input

1 10 100 1000 10000

^d

State Remove

10

List doesn't have the index number : 10

3

5

List doesn't have the index number : 5

4

List doesn't have the index number : 4

^d

[p: 015A5E08, bp: 00FDF8B8, fp: 015A5E40, data:1]

[p: 015A5E40, bp: 015A5E08, fp: 015A4EC0, data:10]

[p: 015A4EC0, bp: 015A5E40, fp: 015A96E8, data:100]

[p: 015A96E8, bp: 015A4EC0, fp: 00FDF8B8, data:10000]

### 【解答例】



## 06-b-04.c

### 【課題】

学籍番号と数学、理科、英語の点数を入力し、学籍番号の昇順、各学生の 3 科目の平均点の昇順で表示するプログラムを作成せよ。

学籍番号に負の数が入力されるまで入力を受け付ける。

データ構造は双方向リストを使用すること。

必要な関数は各自定義して使用してよい。

### 【ヒント】

構造体に番号用、平均点用の fp,bp のペアを定義する。

### 【実行例：入力値は赤字】

Input ID, Math, Science English : **2 100 20 1**

Input ID, Math, Science English : **1 99 99 99**

Input ID, Math, Science English : **3 10 10 10**

Input ID, Math, Science English : **-1 -1 -1 -1**

[ID]

ID : 1, Math : 99, Science : 99, English : 99

ID : 2, Math : 100, Science : 20, English : 1

ID : 3, Math : 10, Science : 10, English : 10

[Average]

Average : 10.000000, ID : 3

Average : 40.333333, ID : 2

Average : 99.000000, ID : 1

### 【解答例】