

Quiz 7-A: Linked Lists

thitichot.k@ku.th สลับบัญชี

 ส่งอีกครั้งเพื่อบันทึก

ระบบจะบันทึกอีเมลของคุณเมื่อส่งแบบฟอร์มนี้

นิสิตต้องตอบคำถามให้กระจ่าง และหากคำตอบมีความคลุมเครือ คำตอบนั้นจะไม่ได้รับคะแนน ในกรณีที่คำถามมีตัวเลือกมาให้ นิสิตต้องเลือกทุกตัวเลือกที่มีส่วนถูกต้อง

ตัว operator ใดที่สามารถใช้เข้าถึงสมาชิกภายในตัวแปร pointer ไปยังข้อมูลประเภท struct

- ☐ address operator (เครื่องหมาย &)
- ☒ arrow operator (เครื่องหมาย ->)
- ☐ dot operator (เครื่องหมาย .)
- ☐ dereferencing operator (เครื่องหมาย *)
- ☐ ไม่มีข้อใดถูก

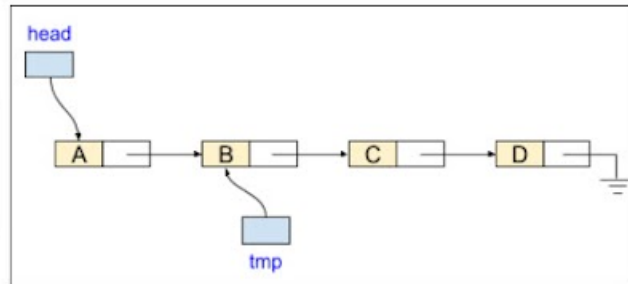
ล้างสิ่งที่เลือก



กำหนดให้โครงสร้าง struct node และตัวแปร head และ tmp ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ จงเลือกใช้คำสั่งที่ให้มาและให้ใช้คำสั่งจำนวนน้อยที่สุดเพื่อทำการลบโหนด B ออกจากลิงก์ลิสต์และคืนหน่วยความจำให้กับระบบ โดยให้พิมพ์คำตอบโดยระบุหมายเลขของคำสั่งที่ให้มาตามลำดับการทำงานและคั่นแต่ละหมายเลขด้วยเครื่องหมาย comma และมีช่องว่าง 1 ช่องหลังเครื่องหมาย comma แต่ละตัว เช่น 1, 2, 3

```
struct node {  
    int data;           // ข้อมูล  
    struct node *next; // node ถัดไป  
} *head, *tmp;
```

```
1 free(head);  
2 head = tmp->next;  
3 head->next = tmp->next;  
4 free(tmp->next);  
5 tmp->next = head->next;  
6 head->next = tmp;  
7 free(tmp);  
8 next = tmp->next;  
9 free(head->next);
```



3, 7

การกำหนดในตัวเลือกใดบ้างที่ไม่ผิดไวยากรณ์ในภาษา C

```
struct node {  
    int data;  
    struct node next;  
};
```

☐ 130413094696

```
typedef struct node {  
    int data;  
    struct node *next;  
} Node;
```

☒ 130433049696

```
struct node {  
    int data;  
    struct node *prev;  
};
```

☒ 130413049696

```
struct node {  
    int data;  
    struct node* prev, next;  
};
```

☐ 130413044696

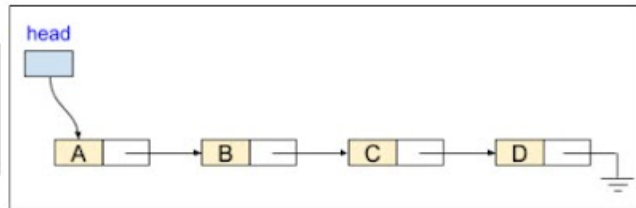
```
typedef struct {  
    int data;  
    struct node *left, *right;  
} Node;
```

☐ 130431049696



กำหนดให้โครงสร้าง struct node และตัวแปร head ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ กำหนดให้ head จะต้องชี้ไปยังโหนดแรกในลิงก์ลิสต์เสมอ จงเลือกตัวเลือกที่เหมาะสมให้กับส่วนของ #A#, #B#, #C#, และ #D# เพื่อให้โค้ดด้านล่างทำการหาผลรวมของข้อมูลจากทุกโหนดที่อยู่ในลิงก์ลิสต์นี้

```
struct node {
    int data;           // ข้อมูล
    struct node *next; // node ถัดไป
} *head;
```



```
1 int main()
2 { int sum=0;
3   struct node *tmp;
4
5   for (#A#; #B#; #C#)
6       sum += #D#;
7 }
```

	#A#	#B#	#C#	#D#
head->data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp = (*tmp)->next	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp->next	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp = tmp->next	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
tmp->data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
head = tmp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp = head	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp != NULL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tmp = head->next	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
head	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*tmp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

tmp = NULL



การทำงานของโปรแกรมด้านล่างนี้เป็นการเพิ่มโหนดเข้าไปใน Circular Doubly Linked List โดยที่หากโหนดถูกเติมอย่างถูกต้องแล้วโปรแกรมควรแสดงข้อมูลออกมาเป็น 1 2 3 ตามที่ถูกเรียงไว้ในลิสต์ จงเลือกตัวเลือก (A), (B), ... (J) จากกล่องสีดำ และเติมลงในบรรทัดที่ถูกเว้นโหนดเอาไว้ เพื่อให้โปรแกรมทำงานตามที่กำหนดไว้

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct node {
5      int data;
6      struct node *prev, *next;
7  } Node;
8
9  int main() {
10     Node dummy, *head, *new_node, *tmp;
11
12     head = &dummy;
13     head->data = 1;
14     head->prev = head->next = head;
15
16     _____;
17     new_node->data = 2;
18     new_node->prev = _____;
19     new_node->next = _____;
20     _____;
21
22     _____;
23     new_node->data = 3;
24     new_node->prev = _____;
25     new_node->next = _____;
26     _____;
27
28     for (tmp=head;; tmp=tmp->next) {
29         printf("%d ", tmp->data);
30         if (tmp == head->prev)
31             break;
32     }
33 }

```

```

(A) tmp = (Node)malloc(sizeof(Node))
(B) tmp = (Node *)malloc(sizeof(Node))
(C) new_node = (Node)malloc(sizeof(Node))
(D) new_node = (Node *)malloc(sizeof(Node))
(E) head
(F) head->prev
(G) head->next
(H) head->prev->next
(I) head->prev->next = head->prev = new_node
(J) head->prev = head->prev->next = new_node

```

(A) (B) (C) (D) (E) (F) (G) (H) (I) (J)

บรรทัด

16 &
22

☐
☐
☐
☒
☐
☐
☐
☐
☐
☐

บรรทัด

18 &
24

☐
☐
☐
☐
☐
☒
☐
☐
☐
☐

บรรทัด

20 &
26

☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☒

บรรทัด

19 &
25

☐
☐
☐
☐
☒
☐
☐
☐
☐
☐
☐

ตัว operator ใดที่สามารถใช้เข้าถึงสมาชิกภายในตัวแปรประเภท struct

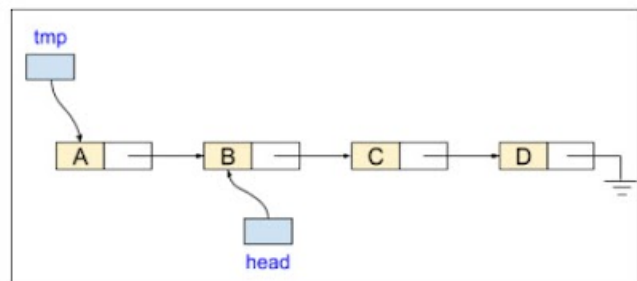
- ☐ arrow operator (เครื่องหมาย ->)
- ☐ address operator (เครื่องหมาย &)
- ☒ dot operator (เครื่องหมาย .)
- ☐ ไม่มีข้อใดถูก
- ☐ dereferencing operator (เครื่องหมาย *)

ล้างสิ่งที่เลือก

กำหนดให้โครงสร้าง struct node และตัวแปร head และ tmp ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ จงเลือกใช้คำสั่งที่ให้มาและให้ใช้คำสั่งจำนวนน้อยที่สุดเพื่อทำการลบโหนด B ออกจากลิงก์ลิสต์และคืนหน่วยความจำให้กับระบบ โดยให้พิมพ์คำตอบโดยระบุหมายเลขของคำสั่งที่ให้มาตามลำดับการทำงานและคืนแต่ละหมายเลขด้วยเครื่องหมาย comma และมีช่องว่าง 1 ช่องหลังเครื่องหมาย comma แต่ละตัว เช่น 1, 2, 3

```
struct node {  
    int data;           // ข้อมูล  
    struct node *next; // node ถัดไป  
} *head, *tmp;
```

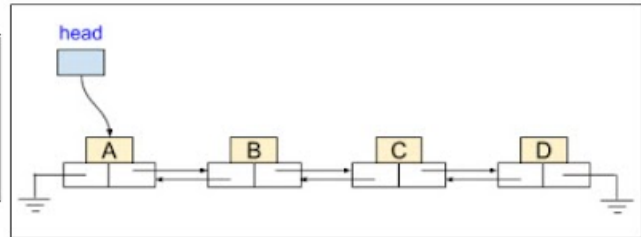
```
1 free(head);  
2 head = tmp->next;  
3 head->next = tmp->next;  
4 free(tmp->next);  
5 tmp->next = head->next;  
6 head->next = tmp;  
7 free(tmp);  
8 next = tmp->next;  
9 free(head->next);
```



5, 1

กำหนดให้โครงสร้าง struct node และตัวแปร head ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ จงระบุนิพจน์ที่สั้นที่สุดที่ใช้เข้าถึงข้อมูล (data) ในโหนด C

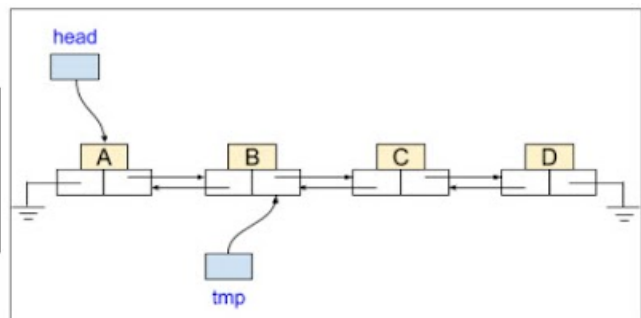
```
struct node {
    int data;           // ข้อมูล
    struct node *prev; // node ก่อนหน้า
    struct node *next; // node ถัดไป
} *head;
```



head->next->next->data

กำหนดให้โครงสร้าง struct node และตัวแปร head และ tmp ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ จงเลือกใช้คำสั่งที่ให้มาและให้ใช้คำสั่งจำนวนน้อยที่สุดเพื่อทำการลบโหนด B ออกจาก doubly linked list และคืนหน่วยความจำให้กับระบบ โดยให้พิมพ์คำตอบโดยระบุหมายเลขของคำสั่งที่ให้มาตามลำดับการทำงานและคั่นแต่ละหมายเลขด้วยเครื่องหมาย comma และมีช่องว่าง 1 ช่องหลังเครื่องหมาย comma แต่ละตัว เช่น 1, 2, 3

```
struct node {
    int data;           // ข้อมูล
    struct node *prev; // node ก่อนหน้า
    struct node *next; // node ถัดไป
} *head, *tmp;
```



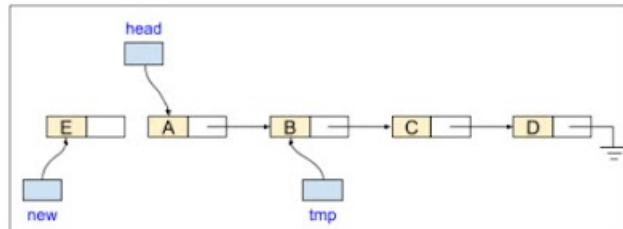
```
1 head->next = tmp->next;
2 free(tmp);
3 head = tmp->next;
4 head->next = tmp->prev;
5 tmp->next->prev = head;
6 free(tmp->next);
7 tmp->next = head->next;
8 tmp->prev->next = tmp;
9 free(head->next);
10 tmp->next->prev = tmp;
11 tmp->prev = head->next;
12 head->next = NULL;
```

1, 5, 2

กำหนดให้โครงสร้าง struct node และตัวแปร head, tmp และ new ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ หากกำหนดให้ head จะชี้ไปยังโหนดแรกในลิงก์ลิสต์เสมอ จงเลือกใช้คำสั่งที่ให้มาและให้ใช้คำสั่งจำนวนน้อยที่สุดเพื่อทำการเพิ่มโหนด E เข้าไปตอนต้นของลิงก์ลิสต์ โดยให้พิมพ์คำตอบโดยระบุหมายเลขของคำสั่งที่ให้มาตามลำดับการทำงาน และคั่นแต่ละหมายเลขด้วยเครื่องหมาย comma และมีช่องว่าง 1 ช่องหลังเครื่องหมาย comma แต่ละตัว เช่น 1, 2, 3

```
struct node {
    int data;           // ข้อมูล
    struct node *next;  // node ถัดไป
} *head, *tmp, *new;
```

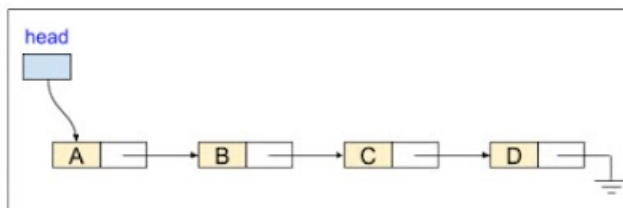
```
1 head->next = tmp;
2 head = tmp->next;
3 head->next = tmp->next;
4 new->next = head;
5 tmp->next = head->next;
6 head = new;
7 head->next = new;
8 next = tmp->next;
9 head = new->next;
```



4, 6

กำหนดให้โครงสร้าง struct node และตัวแปร head ซึ่งมีการประกาศตัวแปรและมีค่าดังแสดงในรูปด้านล่างนี้ จงระบุนิพจน์ที่สั้นที่สุดที่ใช้เข้าถึงข้อมูล (data) ในโหนด D

```
struct node {
    int data;           // ข้อมูล
    struct node *next;  // node ถัดไป
} *head;
```



head->next->next->next->data

กลับ

ส่ง

หน้า 2 จาก 2

ห้ามส่งรหัสผ่านใน Google ฟอรัม

แบบฟอรมนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์ รายงานการละเมิด

Google ฟอรัม



