

알고리즘 스터디 2~3주차 발표

연결리스트 (Linked List)

INDEX

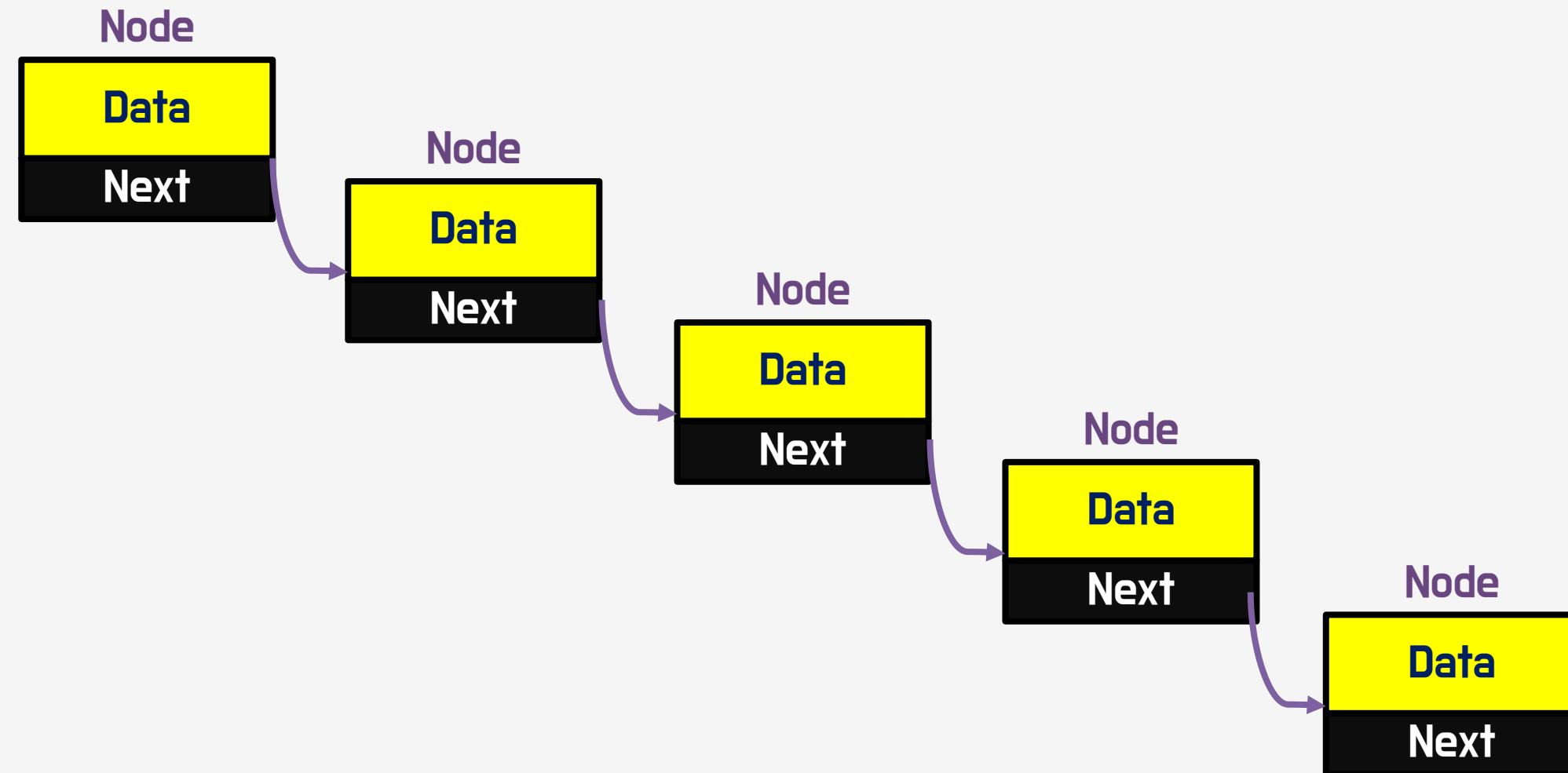
<u>01</u>	연결리스트 소개
<u>02</u>	연결리스트 VS 배열
<u>03</u>	연결리스트 문제
<u>04</u>	-
<u>05</u>	-

CHAPTER.1

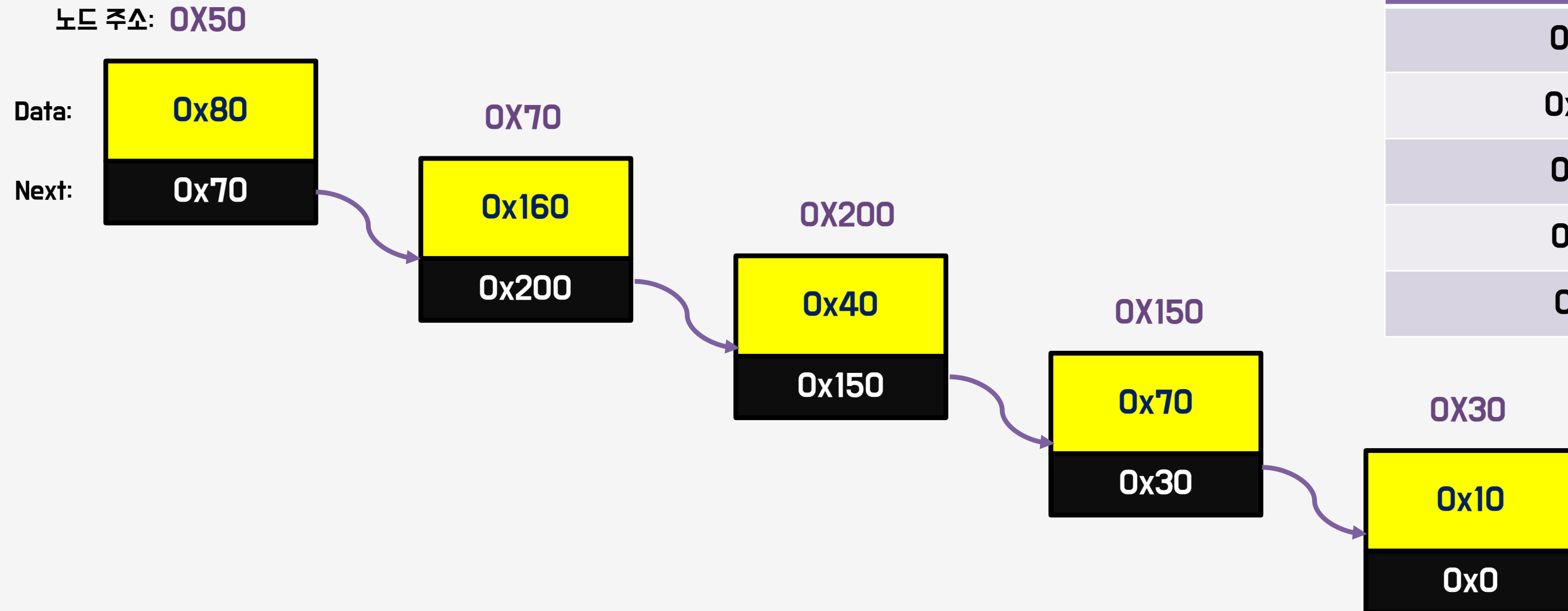
연결리스트 소개

연결리스트 (Linked List)

연결리스트: 노드가 **데이터**와 **포인터**를 가지고 한 줄로 연결돼있는 방식



연결리스트 (Linked List)



문자열 주소	값
0X80	JAVA
0x160	Python
0x40	C++
0x70	GO
0x10	Ruby

```
typedef struct      s_list
{
    struct s_list  *next;
    void           *data;
} t_list;
```

* 리스트의 시작 주소를 담을 변수 필요

* 리스트 마지막 요소의 NEXT => 0

CHAPTER.2

연결리스트 VS 배열

연결리스트 VS 배열

배열

char * argv[] = 0x100

0x100	0x108	0x110	0x118	0x120
0X80	0x160	0x40	0x70	0x10

연결리스트

노드 주소: 0X50

Data:

0x80

Next:

0x70

0X70

0x160

0x200

0X200

0x40

0x150

0X150

0x70

0x30

0X30

0x10

0x0

char * list = 0x50

문자열 주소	값
0X80	JAVA
0x160	Python
0x40	C++
0x70	GO
0x10	Ruby

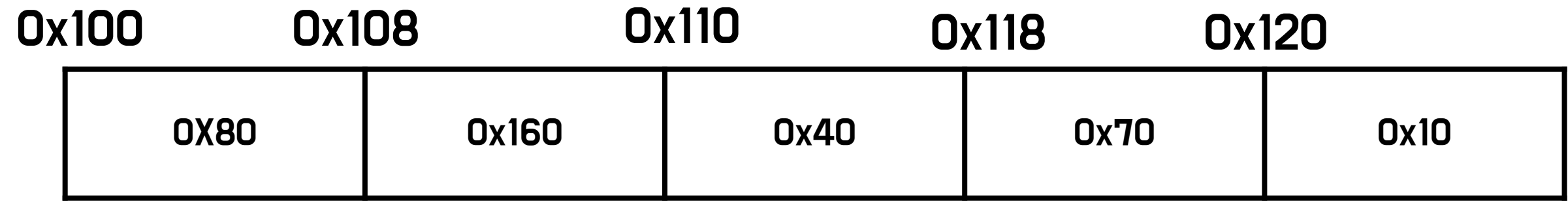
연결리스트 VS 배열 (CASE 1: Indexing)

문자열 주소	값
0X80	JAVA
0x160	Python
0x40	C++
0x70	GO
0x10	Ruby

<3번째 요소를 찾아가기>

배열

char * argv[] = 0x100



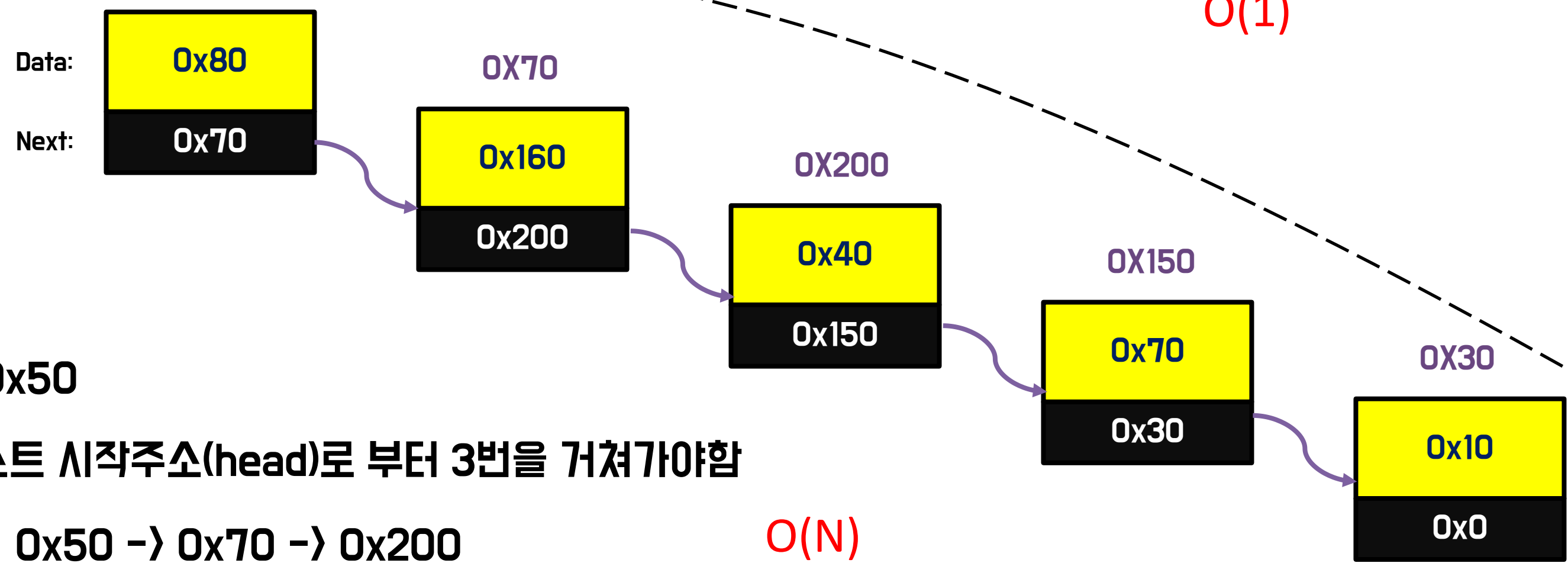
⇒ 그냥 3번째 요소를 바로 찾아가면 됨 (0x110번지)

O(1)

└ 0x100 + 8 * 2

연결리스트

노드 주소: 0X50



char * list = 0x50

⇒ 리스트 시작주소(head)로 부터 3번을 거쳐가야함

0x50 -> 0x70 -> 0x200

O(N)

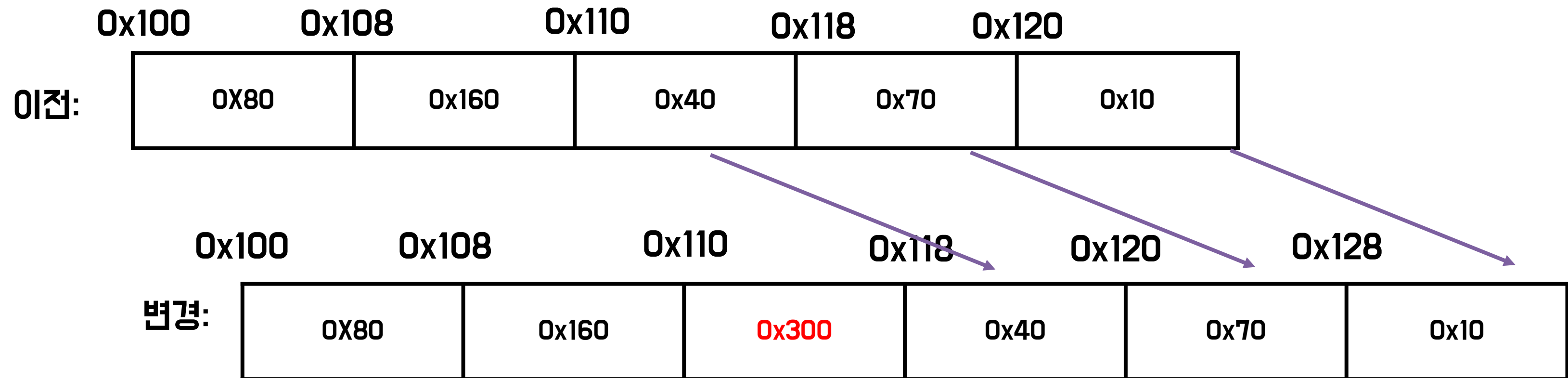
연결리스트 VS 배열 (CASE 2: Insert)

문자열 주소	값
0X80	JAVA
0x160	Python
0x300	C
0x40	C++
0x70	GO
0x10	Ruby

〈3번째 요소에 문자열 추가〉

배열

char * argv[] = 0x100



⇒ 3번째 부터 끝 요소까지 하나씩 뒤로 밀고, 3번째 자리에 값 입력

O(N)

연결리스트 VS 배열 (CASE 2: Insert)

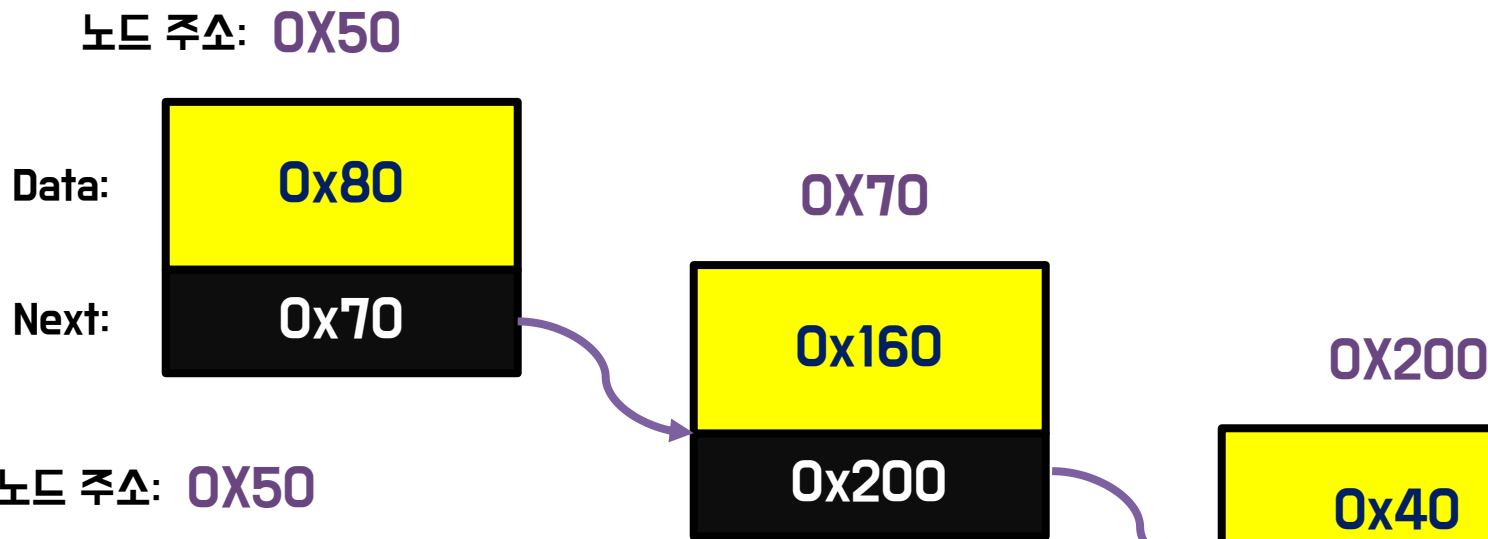
문자열 주소	값
0X80	JAVA
0x160	Python
0x300	C
0x40	C++
0x70	GO
0x10	Ruby

<3번째 요소에 문자열 추가>

char * list = 0x50

연결리스트

이전:

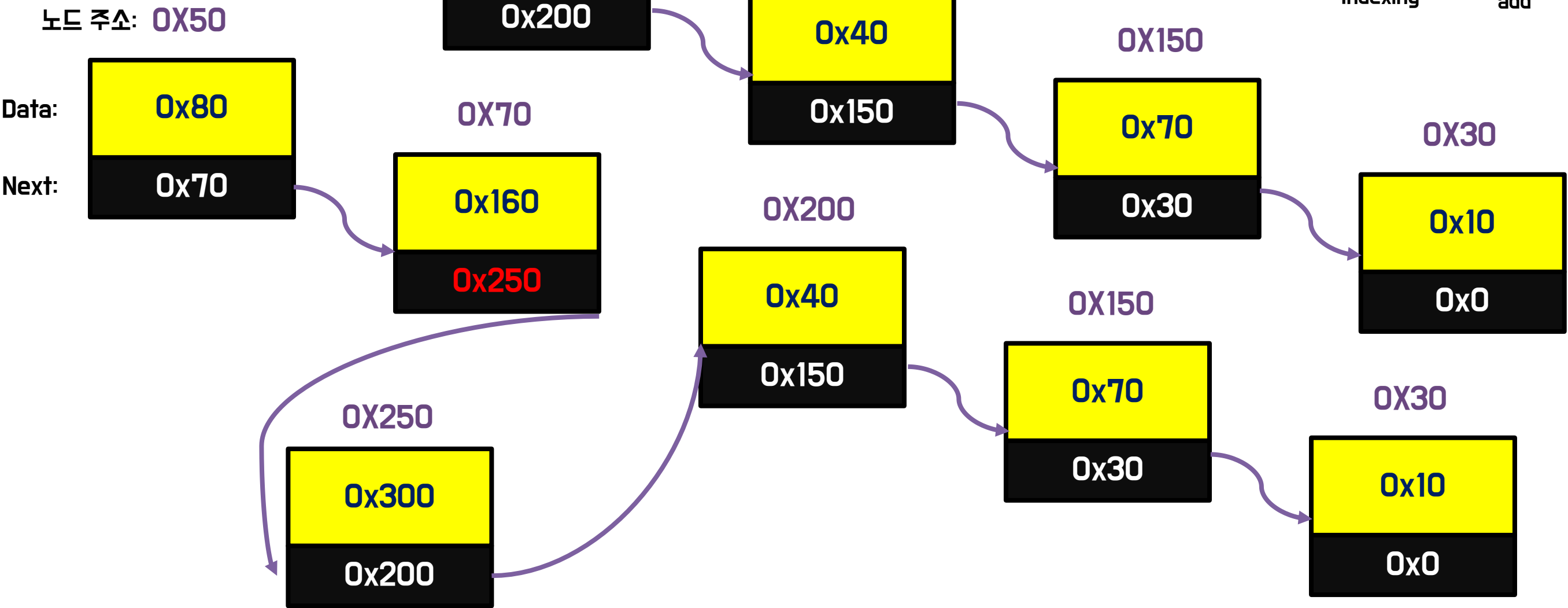


⇒ 3번째까지 찾아가서 노드를 추가해 연결해줌

$O(N) + O(1) = O(N)$

Indexing add

변경:

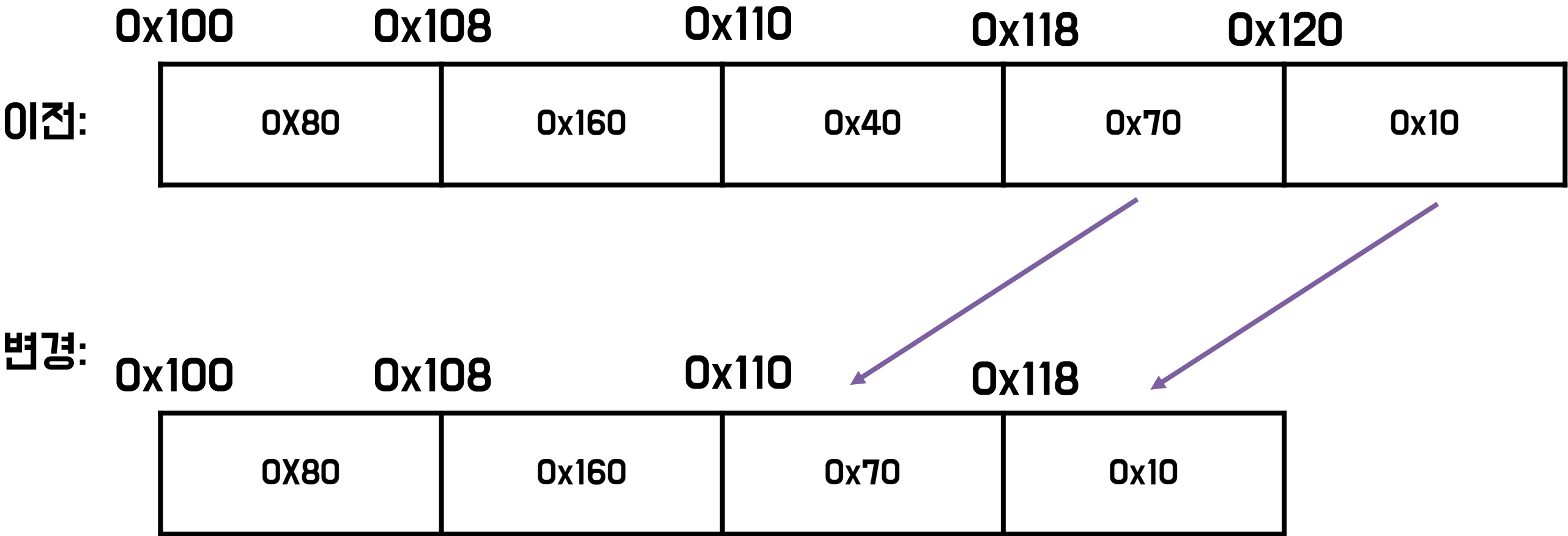


연결리스트 VS 배열 (CASE 3: Delete)

문자열 주소	값
0X80	JAVA
0x160	Python
0x40	C++
0x70	GO
0x10	Ruby

〈3번째 요소를 지우기〉

배열 char * argv[] = 0x100



⇒ 3번째 요소를 삭제 후 뒤에 있는 원소들을 한칸씩 땡긴다.

O(N)

연결리스트 VS 배열 (CASE 3: Delete)

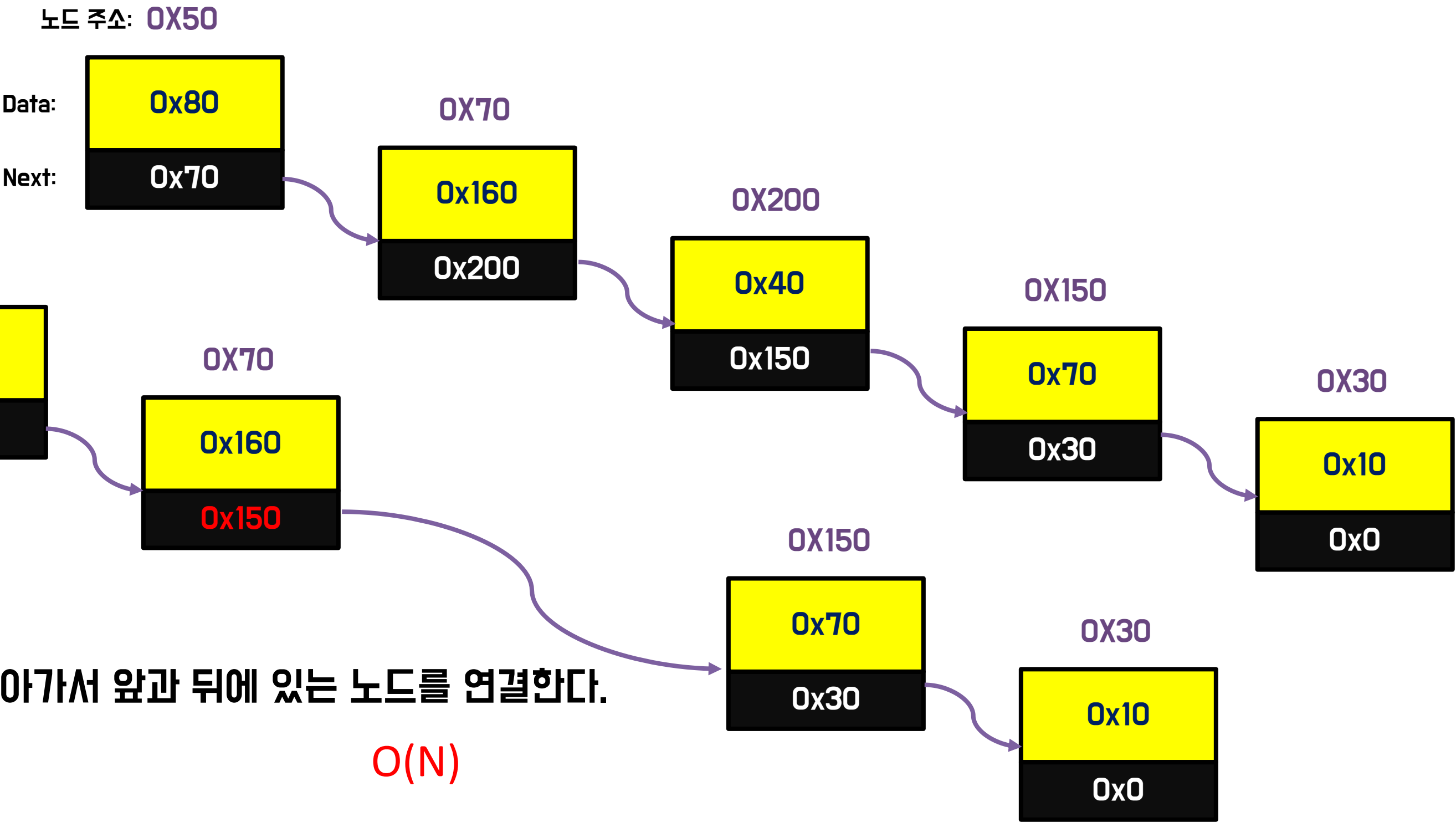
문자열 주소	값
0X80	JAVA
0x160	Python
0x40	C++
0x70	GO
0x10	Ruby

〈3번째 요소를 지우기〉

char * list = 0x50

연결리스트

이전:



⇒ 3번째 요소를 찾아가서 앞과 뒤에 있는 노드를 연결한다.

O(N)

연결리스트 VS 배열 (CASE 3: Delete)

<i>Operations</i>		<i>Array-based implementation</i>	<i>List-based implementation</i>
<i>Common OPs</i>	size, isEmpty	$O(1)$	$O(1)$
<i>Index-based OPs</i>	atIndex, IndexOf, get	$O(1)$	$O(n)$
	set	$O(1)$	$O(n)$
	add, remove	$O(n)$	$O(n)$
<i>Position-based OPs</i>	first, last, prev, next	$O(1)$	$O(1)$
	set	$O(1)$	$O(1)$
	addFirst, addLast	$O(1)$	$O(1)$
	addAfter, addBefore	$O(n)$	$O(1)$
	remove	$O(n)$	$O(1)$

n : the number of elements in the list

CHAPTER.3

연결리스트 문제

연결리스트 문제: 헤더파일

헤더파일

C

```
#include <stdlib.h>
#include <stdio.h>
typedef struct s_list
{
    struct s_list *next;
    void          *data;
}t_list;
```

연결리스트 문제 1: 노드 생성

```
#include "list.h"

t_list *create_elem(void *data)
{
    t_list *temp;

    temp = (t_list *)malloc(sizeof(t_list)); //Node 1개 메모리 할당
    temp->data = data; //데이터 설정
    temp->next = NULL; //다음위치 NULL로 설정
    return (temp);
}
```


연결리스트 문제 1: 노드 생성

```
#include "list.h"

t_list *create_elem(void *data)
{
    t_list *temp;

    temp = (t_list *)malloc(sizeof(t_list)); //Node 1개 메모리 할당
    temp->data = data; //데이터 설정
    temp->next = NULL; //다음위치 NULL로 설정
    return (temp);
}
```

연결리스트 문제 2: list_push_front

```
#include "list.h"

void list_push_front(t_list **begin_list, void *data)
{
    t_list *temp;

    temp = create_elem(data); // Node 생성
    temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
    *begin_list = temp; // list 시작위치를 새로운 노드로 변경
}
```

연결리스트 문제 2: list_push_front

※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```

노드 주소: 0X50

Data:

0x80

Next:

0x00

<변수 표>

변수명	주소	값	*(값)
list	0xFF8	0x50	

<문자열 표>

주소	값
0x80	PYTHON

연결리스트 문제 2: list_push_front

※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```

노드 주소: 0X50



〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x50	
begin_list	0xFF0	0xFFF	0x50
data	0xFE8	0x200	'J'
temp	0xFE0	0x???	

〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

연결리스트 문제 2: list_push_front

※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```

노드 주소: 0X50

Data: 0x80

Next: 0x00

〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

: 코드

: 힙

: 스택

〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x50	
begin_list	0xFF0	0xFFFF	0x50
data	0xFE8	0x200	'J'
temp	0xFE0	0x???	

연결리스트 문제 2: list_push_front

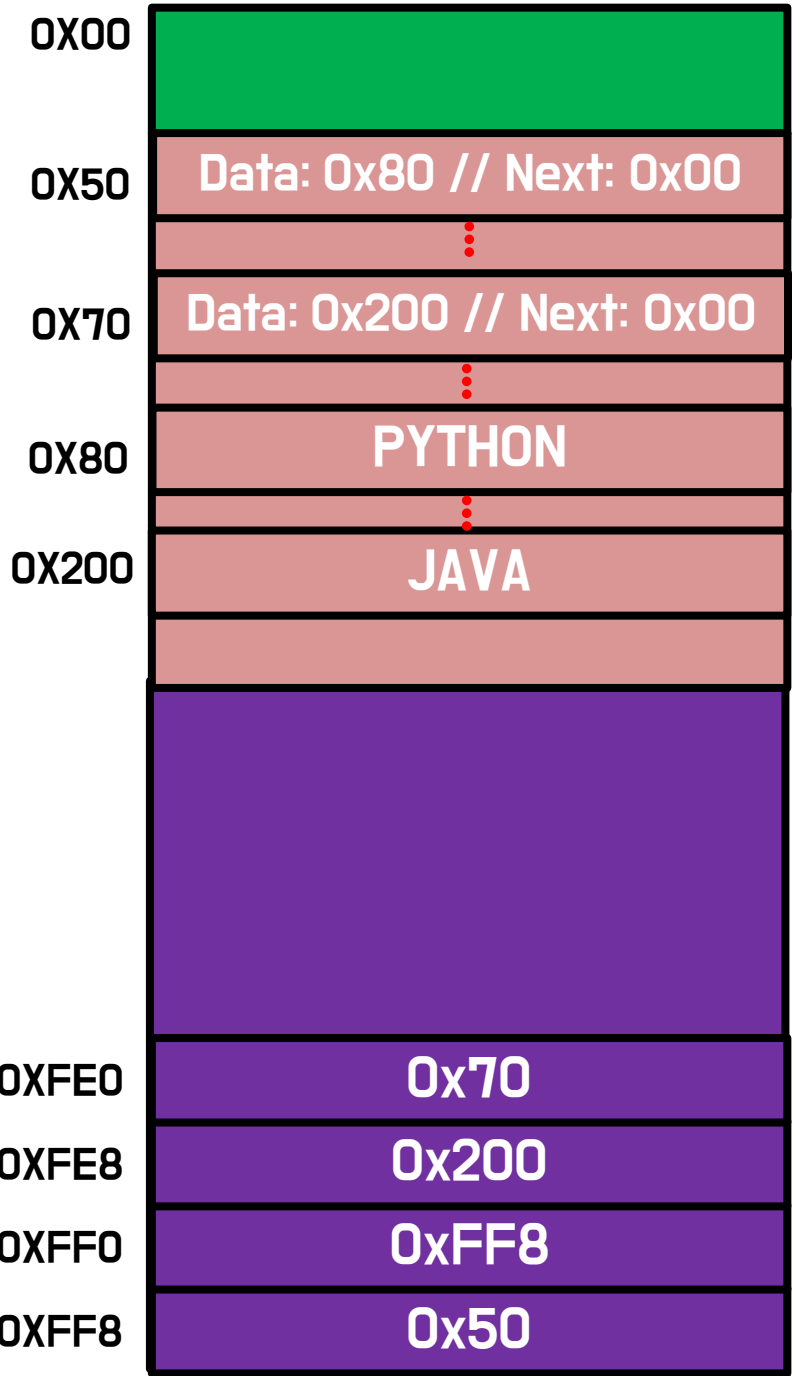
※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```



노드 주소:	0X50	0X70
Data:	0x80	0x200
Next:	0x00	0x00

〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

- : 코드
- : 힙
- : 스택

〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x50	
begin_list	0xFF0	0xFFFF	0x50
data	0xFE8	0x200	'J'
temp	0xFE0	0x0	

연결리스트 문제 2: list_push_front

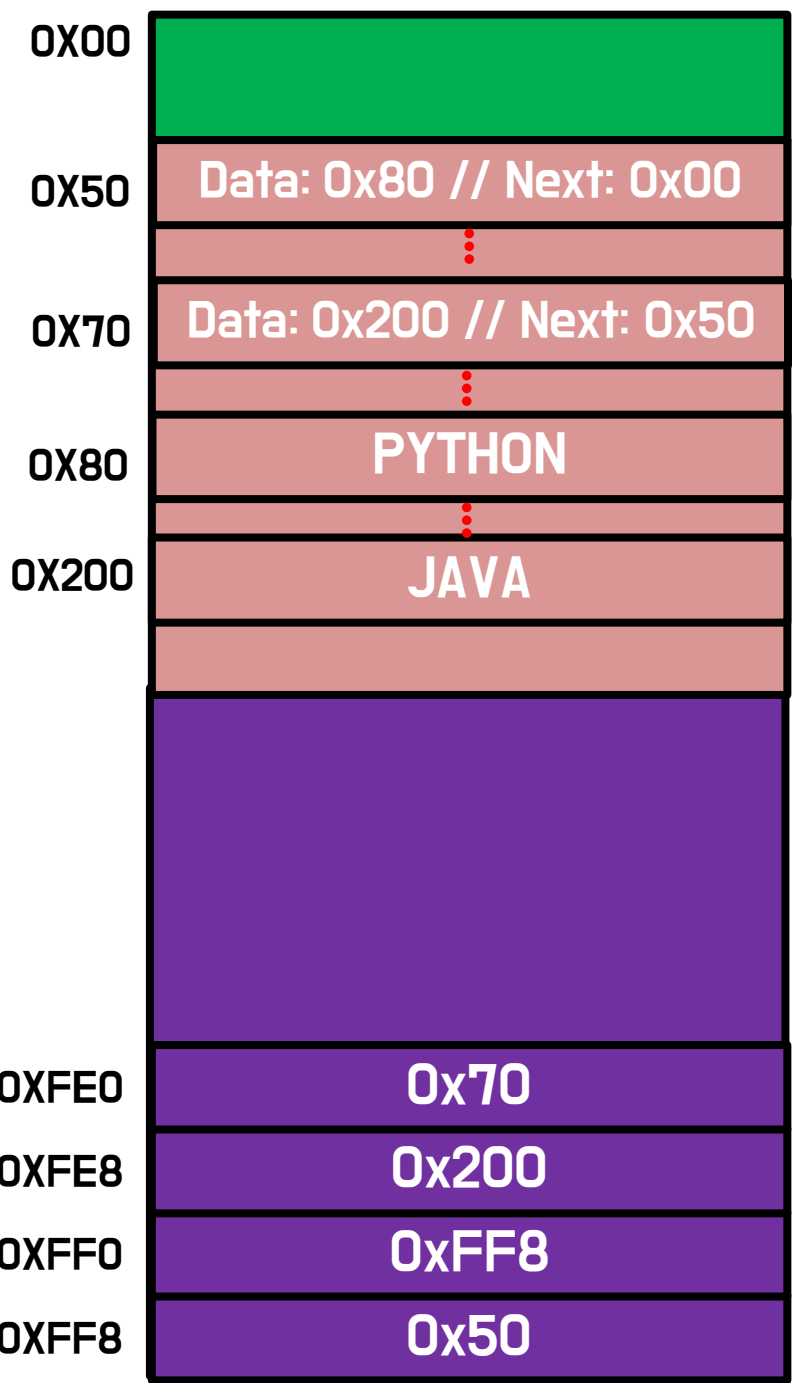
※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```



노드 주소:	0X50	0X70
Data:	0x80	0x200
Next:	0x00	0x50

〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

-  : 코드
-  : 힙
-  : 스택

〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x50	
begin_list	0xFF0	0xFFFF	0x50
data	0xFE8	0x200	'J'
temp	0xFE0	0x0	

연결리스트 문제 2: list_push_front

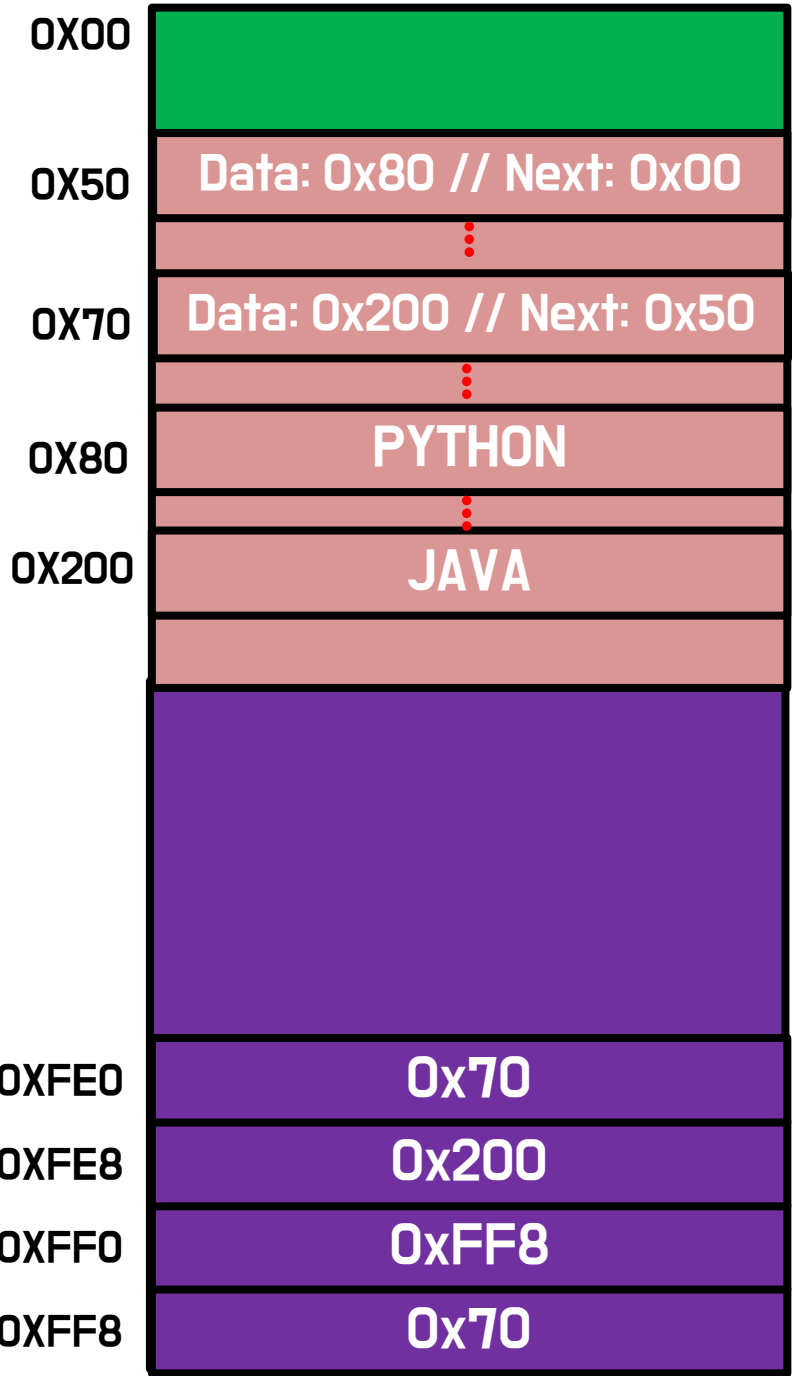
※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```



노드 주소:	0X50	0X70
Data:	0x80	0x200
Next:	0x00	0x50

〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

- : 코드
- : 힙
- : 스택

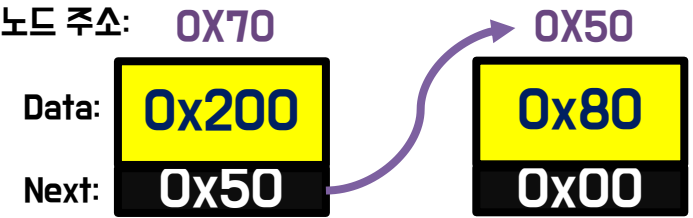
〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x50	
begin_list	0xFF0	0xFFFF	0x50
data	0xFE8	0x200	'J'
temp	0xFE0	0x0	

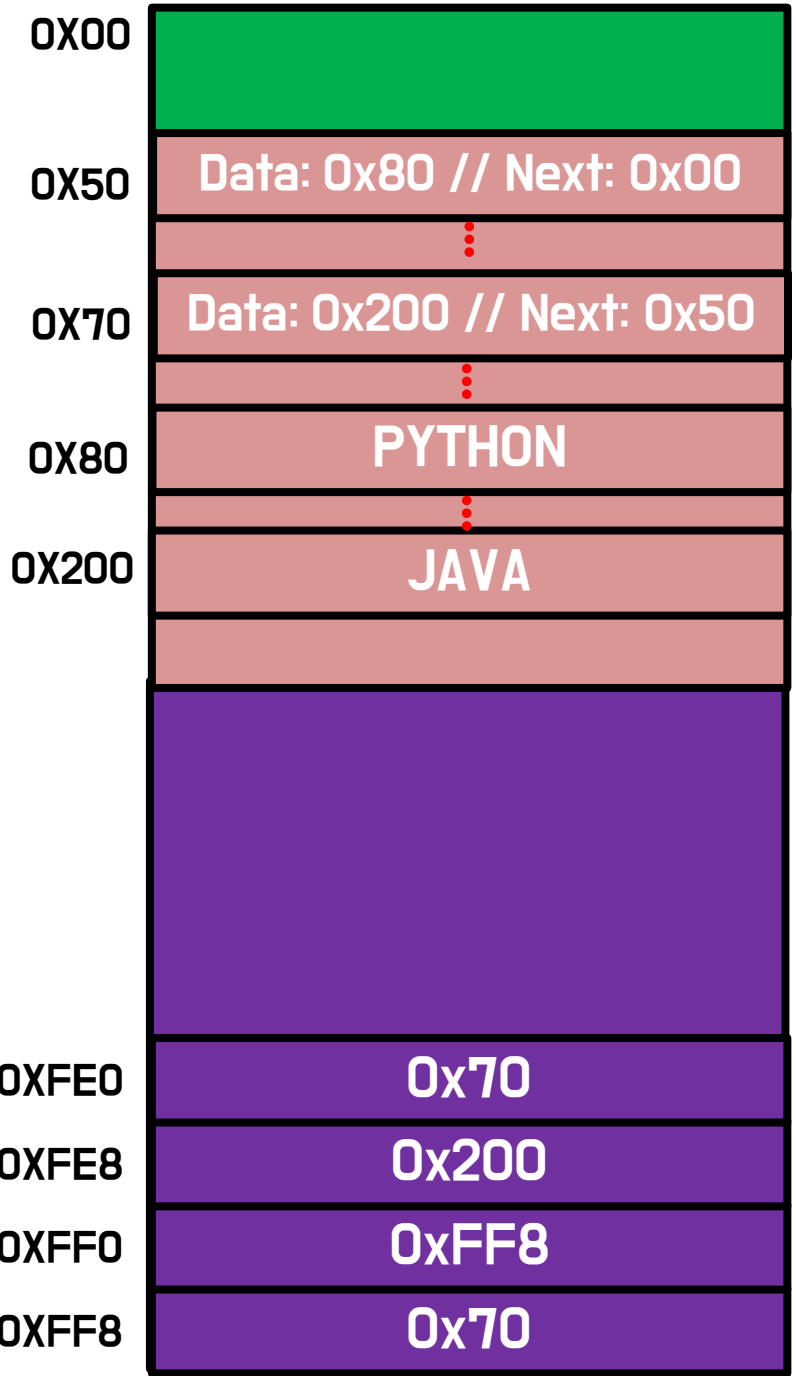
연결리스트 문제 2: list_push_front

※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

: 코드

: 힙

: 스택

〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFFF	0x70
data	0xFE8	0x200	'J'
temp	0xFE0	0x0	

연결리스트 문제 3: list_push_back

```
1  #include "list.h"
2
3  void list_push_back(t_list **begin_list, void *data)
4  {
5      t_list *newe; // 새로운 Node
6      t_list *temp; // 리스트 요소를 가리키는 변수
7
8      newe = create_elem(data); //새로운 Node 생성
9      temp = *begin_list; // 리스트의 시작위치
10     while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11         temp = temp->next; //다음 노드로 이동
12     temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
```

연결리스트 문제 3: list_push_back

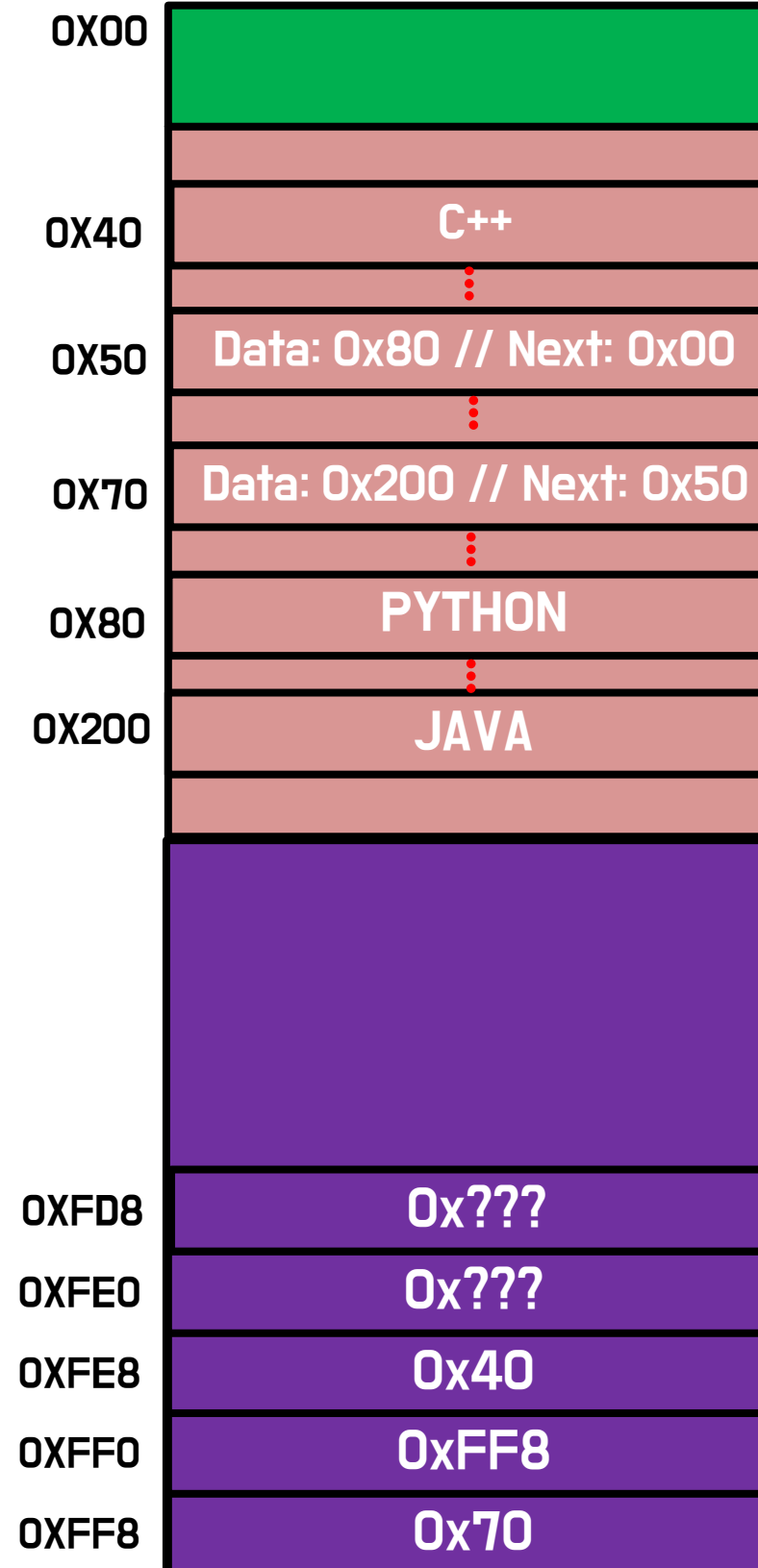
```

1  #include "list.h"
2
3  void list_push_back(t_list **begin_list, void *data)
4  {
5      t_list *newe; // 새로운 Node
6      t_list *temp; // 리스트 요소를 가리키는 변수
7
8      newe = create_elem(data); //새로운 Node 생성
9      temp = *begin_list; // 리스트의 시작위치
10     while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11         temp = temp->next; //다음 노드로 이동
12     temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }

```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++



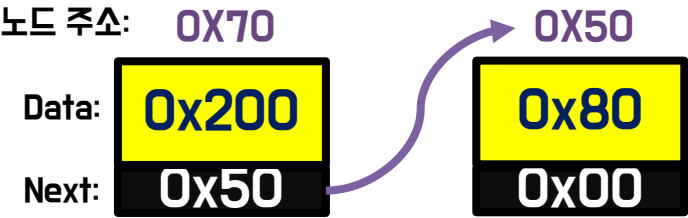
〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x???	
temp	0xFD8	0x???	

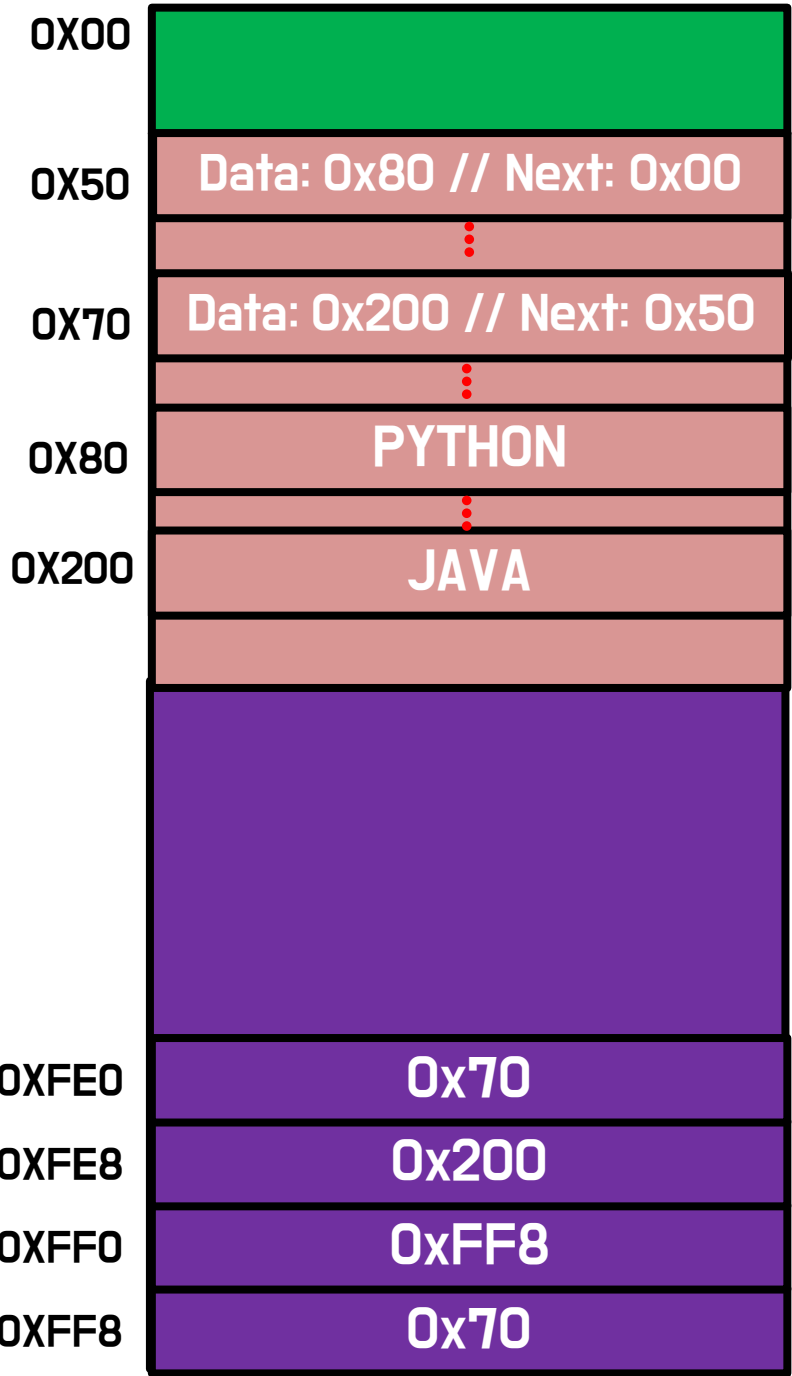
연결리스트 문제 2: list_push_front

※ 테스트 코드

```
1  #include "list.h"
2
3  void list_push_front(t_list **begin_list, void *data)
4  {
5      t_list *temp;
6
7      temp = create_elem(data); // Node 생성
8      temp->next = *begin_list; // 새로운노드의 next를 list의 시작위치로 설정
9      *begin_list = temp; // list 시작위치를 새로운 노드로 변경
10 }
11
12 //테스트코드
13 int main(void)
14 {
15     t_list *list;
16
17     list = create_elem(strdup("PYTHON"));
18     list_push_front(&list, strdup("JAVA"));
19 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA

: 코드

: 힙

: 스택

〈변수 표〉

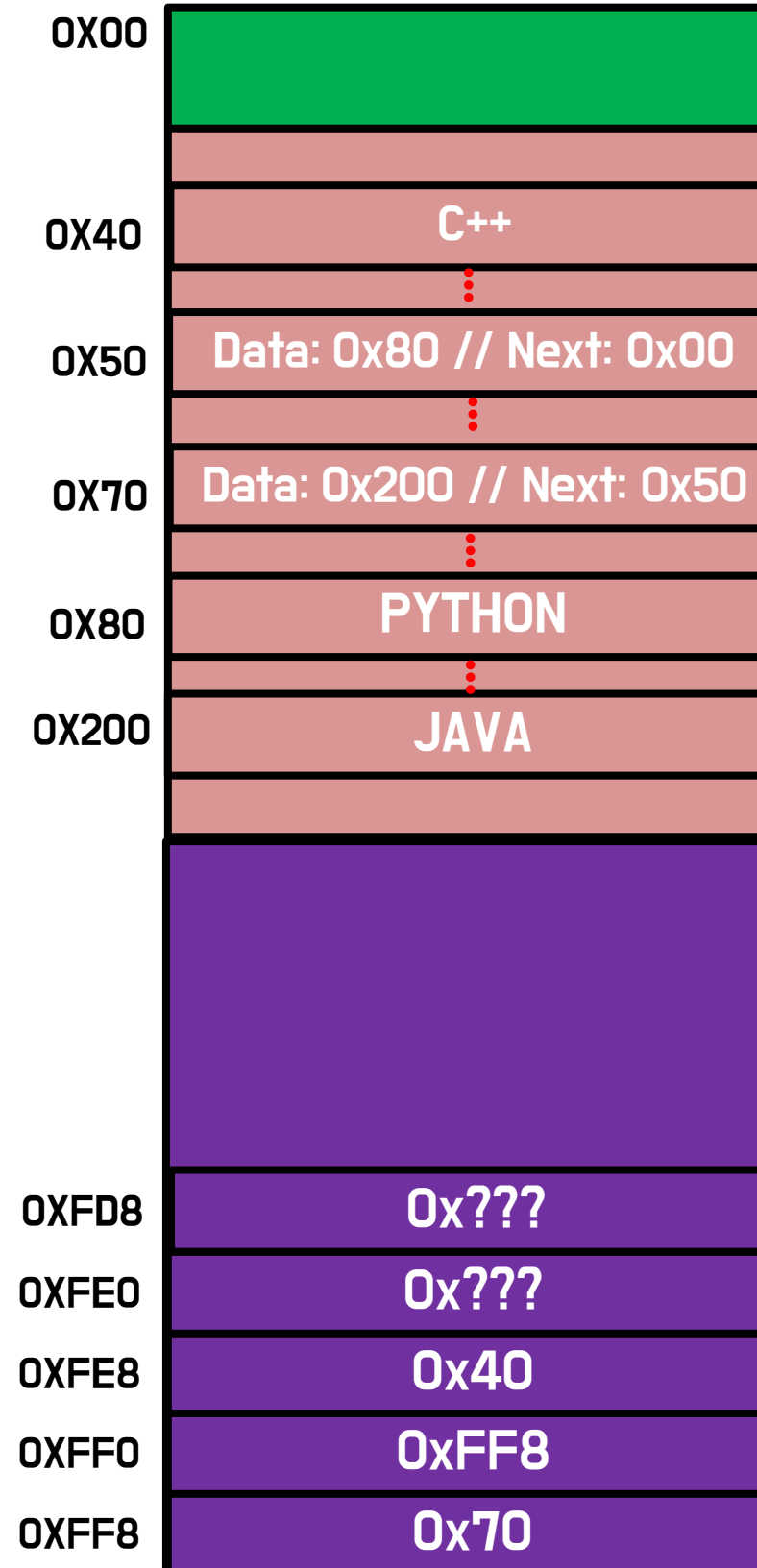
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFFF	0x70
data	0xFE8	0x200	'J'
temp	0xFE0	0x0	

연결리스트 문제 3: list_push_back

```
1  #include "list.h"
2
3  void list_push_back(t_list **begin_list, void *data)
4  {
5      t_list *newe; // 새로운 Node
6      t_list *temp; // 리스트 요소를 가리키는 변수
7
8      newe = create_elem(data); //새로운 Node 생성
9      temp = *begin_list; // 리스트의 시작위치
10     while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11         temp = temp->next; //다음 노드로 이동
12     temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

코드

힙

스택

〈변수 표〉

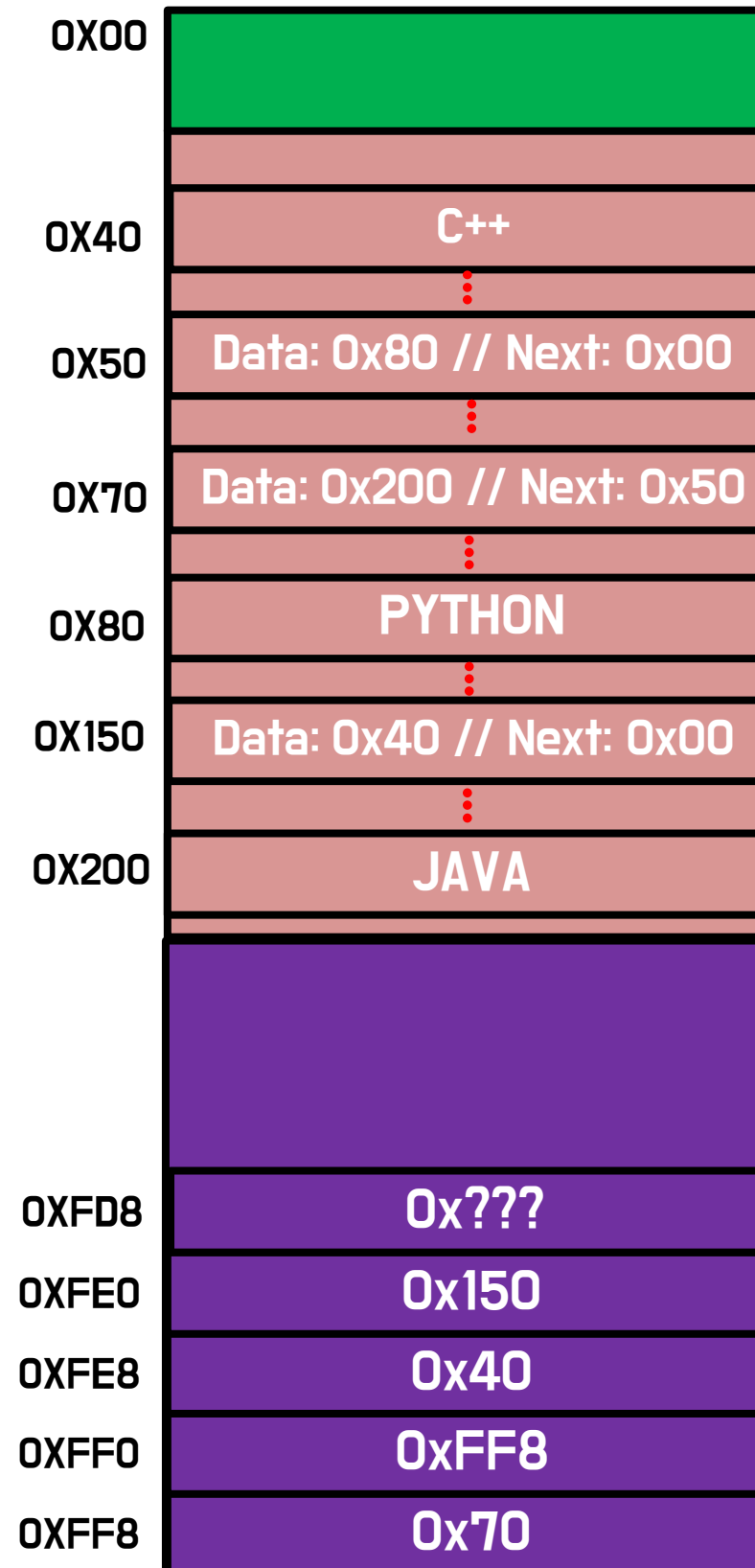
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x???	
temp	0xFD8	0x???	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

: 코드

: 힙

: 스택

〈변수 표〉

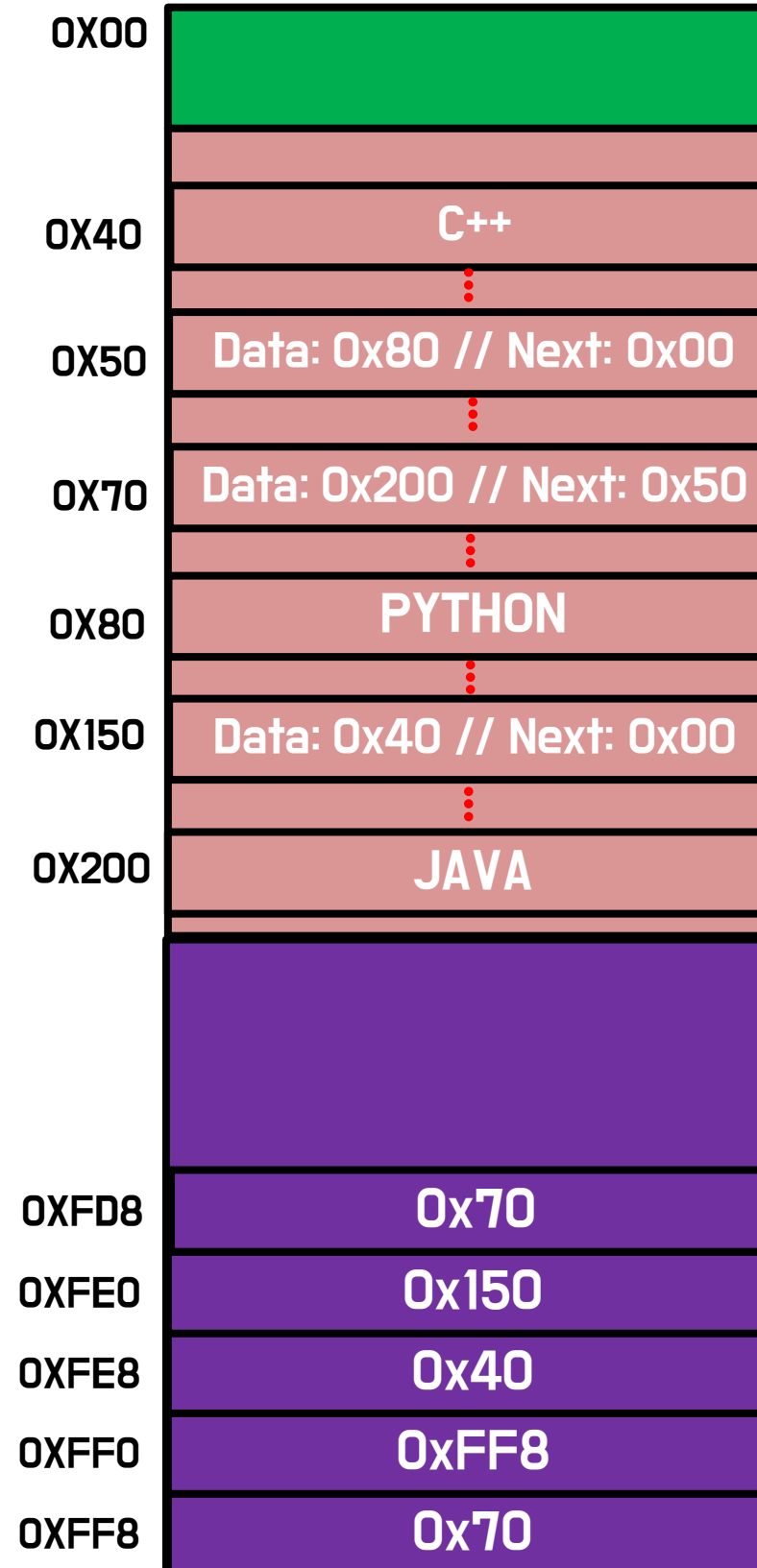
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x???	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

<div></div>	: 코드
<div></div>	: 힙
<div></div>	: 스택

〈변수 표〉

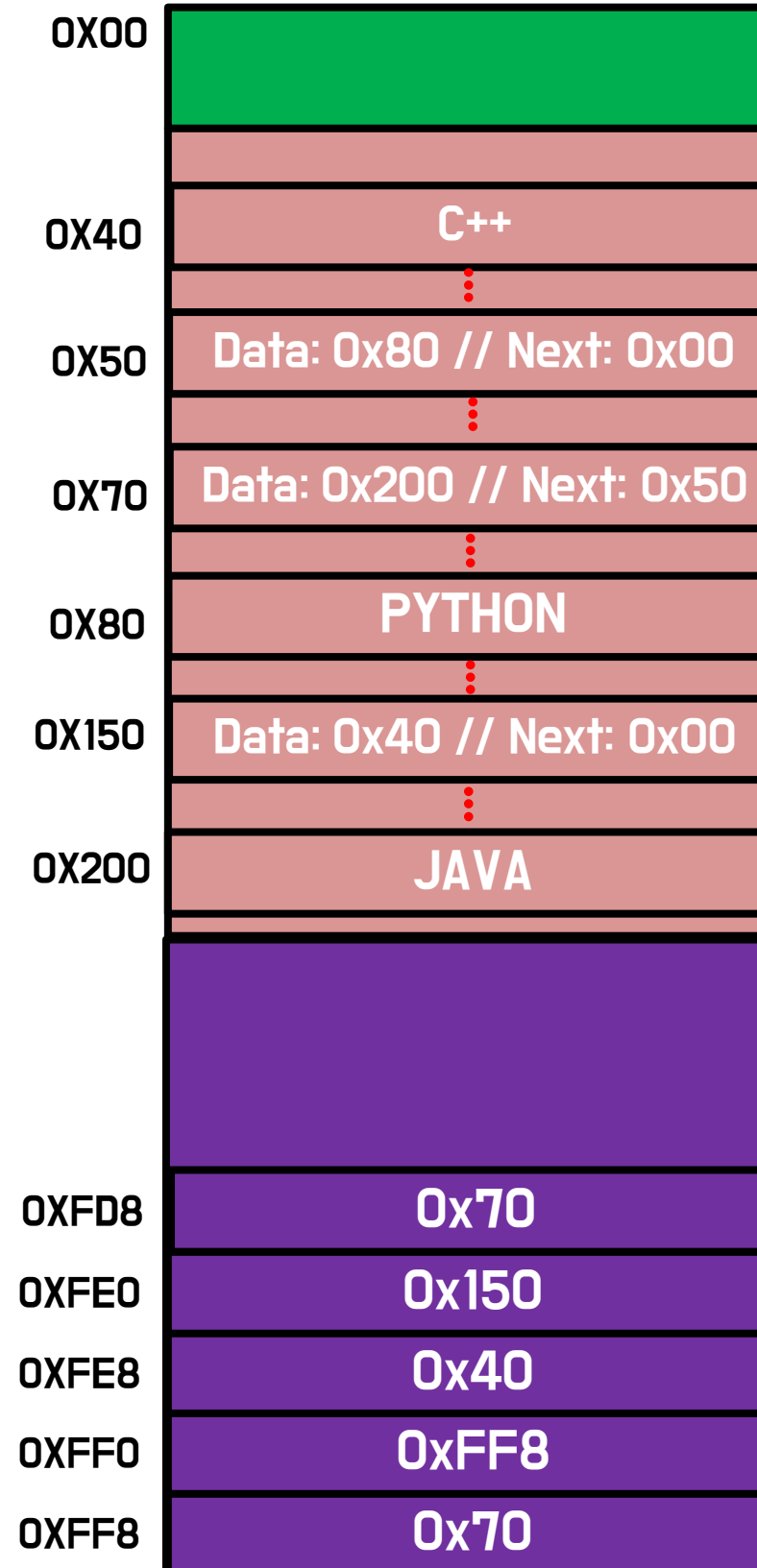
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x70	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

 : 코드

 : 힙

 : 스택

〈변수 표〉

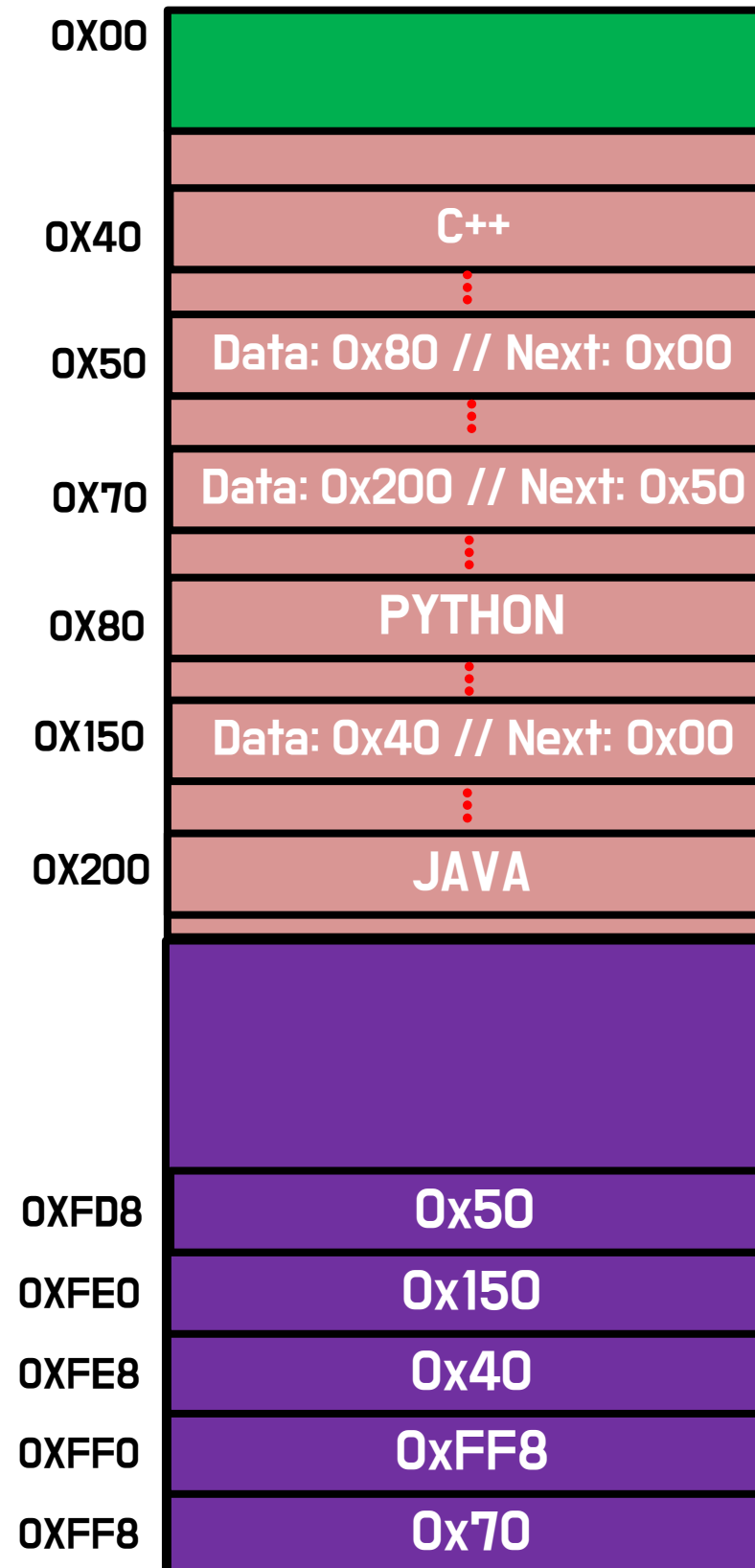
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x70	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

: 코드

: 힙

: 스택

〈변수 표〉

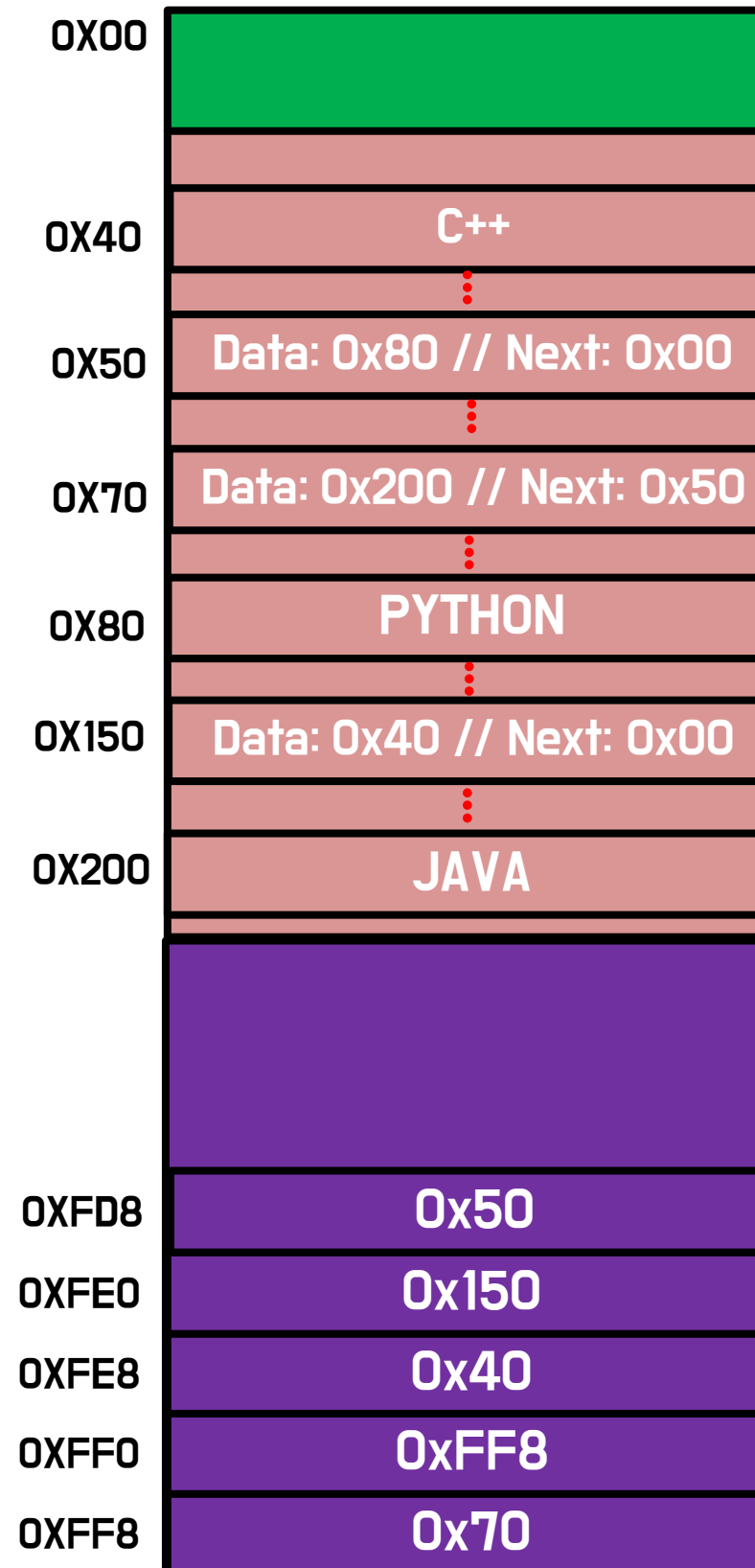
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x50	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

 : 코드

 : 힙

 : 스택

〈변수 표〉

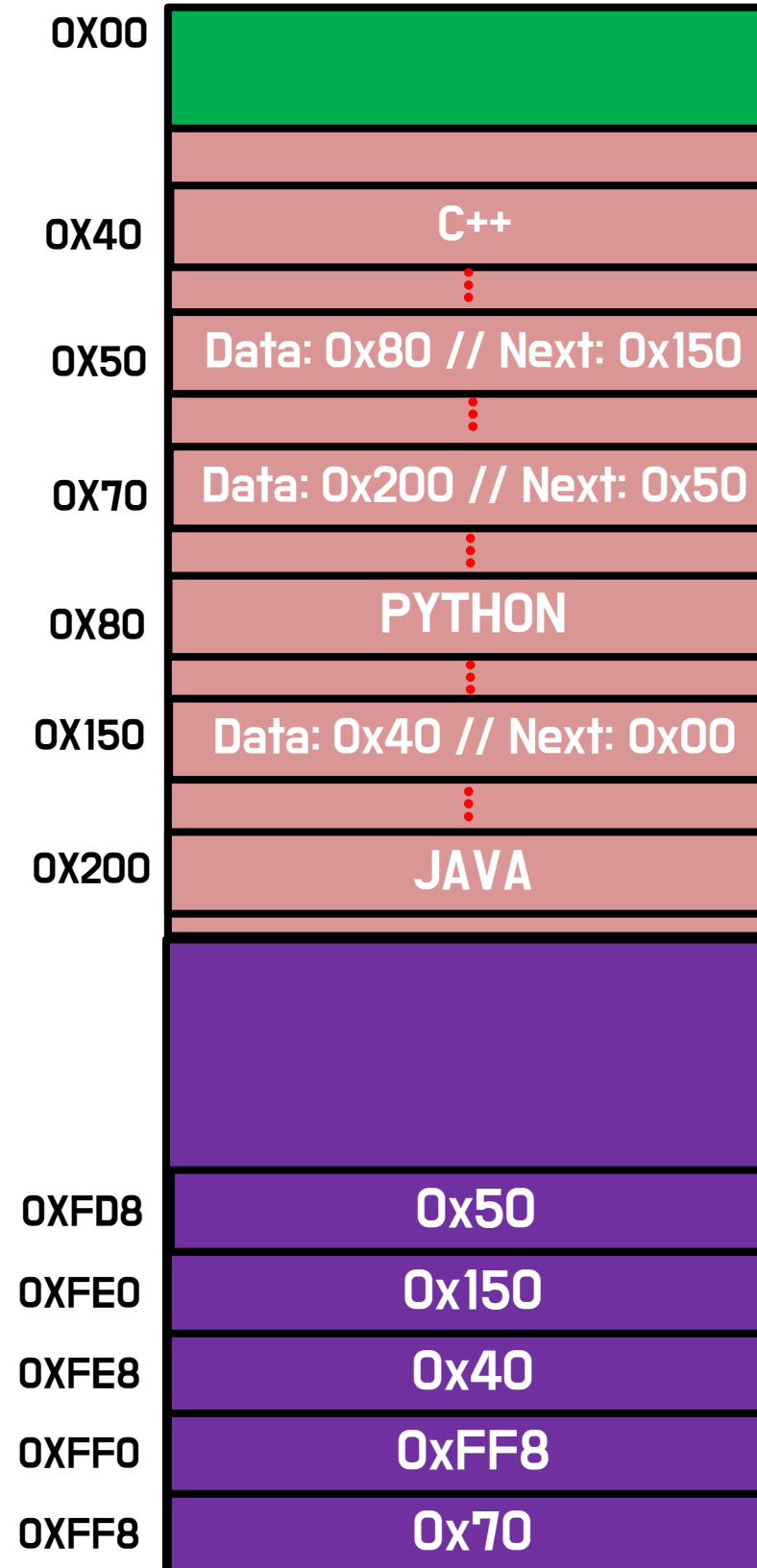
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x50	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

: 코드

: 힙

: 스택

〈변수 표〉

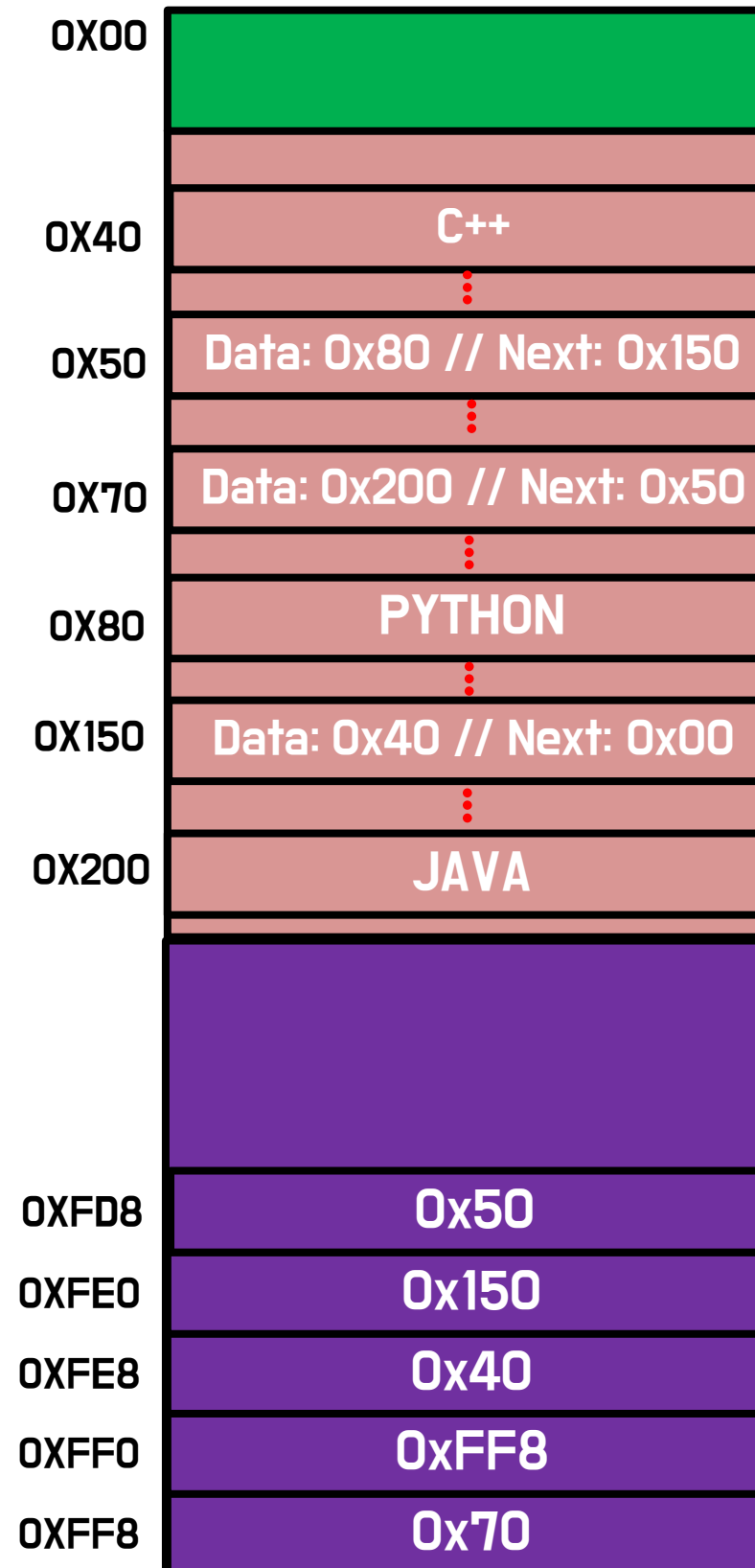
변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x50	

연결리스트 문제 3: list_push_back

```
1 #include "list.h"
2
3 void list_push_back(t_list **begin_list, void *data)
4 {
5     t_list *newe; // 새로운 Node
6     t_list *temp; // 리스트 요소를 가리키는 변수
7
8     newe = create_elem(data); //새로운 Node 생성
9     temp = *begin_list; // 리스트의 시작위치
10    while (temp->next != NULL) // 현재 노드의 다음 노드가 있다면
11        temp = temp->next; //다음 노드로 이동
12    temp->next = newe; //마지막 노드의 다음 노드에 새로 생성한 노드 연결
13 }
14
15 //테스트 코드
16
17 int main(void)
18 {
19     t_list *list;
20
21     list = create_elem(strdup("PYTHON"));
22     list_push_front(&list, strdup("JAVA"));
23     //front 까지의 코드
24
25     list_push_back(&list, strdup("C++"));
26 }
```



〈메모리〉



〈문자열 표〉

주소	값
0x80	PYTHON
0x200	JAVA
0x40	C++

: 코드

: 힙

: 스택

〈변수 표〉

변수명	주소	값	*(값)
list	0xFF8	0x70	
begin_list	0xFF0	0xFFF	0x70
data	0xFE8	0x40	'C'
newe	0xFE0	0x150	
temp	0xFD8	0x50	

CHAPTER.4

—

CHAPTER.5

—

감사합니다