

CS50 코칭스터디 2기

boostcourse 부스트코스

좋아요 6608 수강생 1342

CS50 코칭스터디란? 📖

CS50 2기 공지글 📄

CS50 2기 코치단 👥

2기 부스터 소통공간 🗨️ 📺

개인 미션 공간 🗨️

2기 리드부스터 소통공간 🗨️

2기 부스터 문의사항 🗨️

△ 학습 인증 안내

▼ 미션공개&라이브특강 ▼

0주차_오리엔테이션

1주차_컴퓨팅사고

2주차_C언어

3주차_배열

4주차_알고리즘

5주차_메모리

6주차_자료구조

❤️ 6주차(2월 22일 ~ 2월 26일) ❤️ 부스터여러분
~ 마지막까지 힘내요!!!

○ 📖 자료구조

○ 💎 6주차 미션(2월 25일 목요일 오후 ...

🔒 코딩은 랜션을 타고

Alice코치 (team1~10)

💎 6주차 미션(2월 25일 목요일 오후 6시 마감)

📅 6주차 미션

💡 샘플미션 (제출용X)

▶ 부스터 여러분들께서 미션을 수행하시는데 도움이 될 수 있게 미션샘플을 답지와 함께 제공해드립니다. 미션샘플은 제출용이
래의 미션을 풀기위해 참고하기 위한 미션입니다.

✓ 샘플미션.

1. 미션 제목

2개의 리스트 합치기

2. 지시문

EDWITH CS50 강좌를 모두 수강한 여러분은 유능한 개발자로 회사에 소문이나 핵심 부서로 배치되었습니다. 핵심부서의 주요
하나는 다른 부서의 업무를 중합하는 일입니다. 부서 배치 첫 업무로 A부서에서 수행한 업무와 B부서에서 수행한 업무를 합치는
게 되었습니다.A부서에서는 미국 지사들의 매출이 오름차순으로 정렬된 자료를 연결리스트 형태로 보내왔고, B부서에서는 한국 지사들의 매출
차순으로 정렬된 자료를 연결리스트 형태로 보내왔습니다.여러분의 업무는 이제 두 연결리스트를 하나의 연결리스트로 합치어 한국과 미국 지사들의 매출이 오름차순으로 정렬된 연결리
스트를 만드는 것입니다. 예를 들어, A부서에서 보내온 연결리스트가 2->6->9->10, B부서에서 보내온 연결리스트가 1->5->7->8-
라면 여러분이 만들 연결리스트는 1->2->5->6->7->8->9->10->11 이 되어야 합니다.

업무 수행을 위해 연결 리스트를 합치는 함수와 합쳐진 연결 리스트를 출력하는 함수를 만들어주세요.

```

1  /**
2   * 연결 리스트의 형태는 아래와 같습니다.
3   * struct ListNode {
4   *     int sales;
5   *     struct ListNode *next;
6   * } ListNode;
7   */
8
9
10 struct ListNode* mergeTwoLists(ListNode* list1, ListNode* list2){
11     // 함수를 채워주세요!
12 }
13
14 void append(ListNode* l, int data) {
15     // 함수를 채워주세요.
16 void printMergedLinkedList(ListNode* list3) {
17     // 함수를 채워주세요!
18 }
19
20
21 int main() {
22     ListNode* listA = (ListNode*)malloc(sizeof(ListNode));
23     ListNode* listB = (ListNode*)malloc(sizeof(ListNode));

```

<

>

수강완료

L 코치질문

L 미션제출

수료증

성적조회

```

26  append(listA, 6);
27  append(listA, 9);
28  append(listA, 10);
29  printList(listA);
30
31  append(listB, 1);
32  append(listB, 5);
33  append(listB, 7);
34  append(listB, 8);
35  append(listB, 11);
36  printList(listB);
37
38  ListNode* result = mergeTwoLists(listA, listB);
39  printList(result);
40  }

```

main 함수에는 위와 같이 정렬된 연결 리스트 2개가 미리 만들어져 있다고 가정합니다. 여러분이 작성하신 함수 mergeTwoLists append 함수 그리고 printMergedLinkedList를 순서대로 호출하면 정렬된 결과 출력이 나오도록 작성해 주세요.

A 부서와 B 부서에서 보내는 연결리스트는 모두 오름차순으로 정렬되어 있는 상태이며, A 부서에서 보내는 리스트와 B 부서에서 온 리스트의 길이는 다를 수 있습니다.

입력값: 정렬된 2개의 연결리스트(Main 함수에 문제와 같이 선언되어 있다고 가정)
출력값: 정렬된 리스트가 출력되게 해주세요.

위의 문제라면, 다음과 같이 출력 값이 나와야 합니다.

```

2 6 9 10
1 5 7 8 11
1 2 5 6 7 8 9 10 11

```

3. 핵심 개념

#연결리스트 #리스트 삽입 #리스트 합치기 #리스트 출력 #병합정렬

🔔 답안

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct node {
5      int data;
6      struct node* next;
7  }ListNode;
8
9  ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
10     ListNode* head = (ListNode *)malloc(sizeof(ListNode));
11     ListNode* pt = head;
12
13     l1 = l1->next;
14     l2 = l2->next;
15
16     while (l1!=NULL && l2!=NULL) {
17         if (l1->data <= l2->data) {
18             pt->next = l1;
19             l1 = l1->next;
20         } else {
21             pt->next = l2;
22             l2 = l2->next;
23         }
24     }
25     pt->next = l1;

```



수강완료

```

29     pt->next = l1;
30     if (l2 != NULL)
31         pt->next = l2;
32     return head;
33 }
34
35 void append(ListNode* l, int data) {
36     ListNode* item = (ListNode*)malloc(sizeof(ListNode));
37     item->data = data;
38     item->next = NULL;
39
40     ListNode* temp = (ListNode*)malloc(sizeof(ListNode));
41     temp = l;
42     while(temp->next != NULL) {
43         temp=temp->next;
44     }
45     temp->next = item;
46 }
47
48 void printList(ListNode* l) {
49     while(l->next != NULL) {
50         printf("%d ", l->next->data);
51         l = l->next;
52     }
53     printf("\n");
54 }
55
56
57 int main() {
58     ListNode* listA = (ListNode*)malloc(sizeof(ListNode));
59     ListNode* listB = (ListNode*)malloc(sizeof(ListNode));
60
61     append(listA, 2);
62     append(listA, 6);
63     append(listA, 9);
64     append(listA, 10);
65     printList(listA);
66
67     append(listB, 1);
68     append(listB, 5);
69     append(listB, 7);
70     append(listB, 8);
71     append(listB, 11);
72     printList(listB);
73
74     ListNode* result = mergeTwoLists(listA, listB);
75     printList(result);
76 }

```

✓ 미션 1.

1. 미션 제목
배열로 Stack 만들기

2. 지시문



수강완료

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct stack{
5      int top;
6      int capacity;
7      int* array;
8  } Stack;
9
10 Stack* createStack(int capacity) {
11     Stack* stack = (Stack*)malloc(sizeof(Stack));
12     stack->capacity = capacity;
13     stack->top = -1;
14     stack->array = (int *)malloc(stack->capacity*sizeof(int));
15     return stack;
16 }
17
18 int isFull(Stack* stack) {
19     return stack->top == stack->capacity-1;
20 }
21
22 int isEmpty(Stack* stack) {
23     return stack->top == -1;
24 }
25
26 void push(Stack* stack, int item) {
27     if (isFull(stack))
28         return;
29     stack->array[++stack->top] = item;
30     printf("%d pushed to stack\n", item);
31 }
32
33 int pop(Stack* stack) {
34     // 이곳을 채워주세요!
35 }
36
37 int peek(Stack* stack) {
38     // 이곳을 채워주세요!
39 }
40
41 int main() {
42     Stack* stack = createStack(100);
43
44     push(stack, 10);
45     push(stack, 20);
46     push(stack, 30);
47     push(stack, 40);
48
49     printf("%d pop from stack\n", pop(stack));
50     printf("%d pop from stack\n", pop(stack));
51
52     push(stack, 50);
53     printf("%d pop from stack\n", pop(stack));
54     printf("%d pop from stack\n", pop(stack));
55     printf("%d pop from stack\n", pop(stack));
56     printf("%d pop from stack\n", pop(stack));
57     return 0;
58 }
```

```

20 pushed to stack
30 pushed to stack
40 pushed to stack
40 pop from stack
30 pop from stack
50 pushed to stack
50 pop from stack
20 pop from stack
10 pop from stack
-9999 pop from stack

```

위와 같은 결과가 나오도록 작성해 주세요. 다양한 숫자와 사례를 만들어서 실험해 보세요!

3. 핵심 개념

#Stack #배열 #Stack 구현

✓ 미션 2.

1. 미션 제목

연결리스트로 Stack 만들기

2. 지시문

EDWITH CS50 강좌에서 배운 Stack을 보조미션 1에서 배열을 이용해서 구현해 보셨는데요, 이번에는 연결리스트를 이용해서 구현하는 과제입니다. 지난 문제와 마찬가지로 아래 표에 함수의 주석 처리된 부분들에 여러분의 코드를 채워 넣어주세요.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct stackNode {
5      int data;
6      struct stackNode* next;
7  } StackNode;
8
9  StackNode* createStackNode(int data) {
10     StackNode* node = (StackNode*)malloc(sizeof(StackNode));
11     node->data = data;
12     node->next = NULL;
13     return node;
14 }
15
16 int isEmpty(StackNode* root) {
17     return !root;
18 }
19
20 void push(StackNode** root, int data) {
21     // 이 부분을 구현해 주세요!
22     printf("%d pushed to stack\n", data);
23 }
24
25 int pop(StackNode** root) {
26     if (isEmpty(*root))
27         return -9999;
28     int popped;
29     // 이 부분을 구현해 주세요!
30     return popped;
31 }
32
33 int peek(StackNode** root) {
34     if (isEmpty(*root))
35         return -9999;

```



수강완료

```

39 }
40
41 int main() {
42     StackNode* root = NULL;
43
44     push(&root, 10);
45     push(&root, 20);
46     push(&root, 30);
47     push(&root, 40);
48
49     printf("%d pop from stack\n", pop(&root));
50     printf("%d pop from stack\n", pop(&root));
51
52     push(&root, 50);
53     printf("%d peeked from stack\n", peek(&root));
54     printf("%d pop from stack\n", pop(&root));
55     printf("%d pop from stack\n", pop(&root));
56     printf("%d pop from stack\n", pop(&root));
57     return 0;
58 }

```

Main 함수를 실행시키면 Stack 출력 결과가 정상적으로 나와야 합니다. 배열과 연결리스트를 이용해서 구현을 해 보셨는데요, (이 더 마음에 드셨나요? Main 함수의 내용을 수정해 보면서 실험도 해보시고, Stack 을 구현할 때 리스트를 사용한 방식과 배열 한 방식 중 어떤 방식이 더 좋았는지 느낀 점도 공유해 주세요!

3. 핵심 개념

#Stack #연결리스트 #Stack 구현

✓ 미션 3.

1. 미션 제목

배열로 Queue 만들어보기!

2. 지시문

이번 과제에서는 Queue를 구현해 봅시다! Stack 과 Queue의 구현은 얼핏 보면 비슷해 보이지만 막상 구현해 보면 많은 부분이 다릅니다. 어떻게 구현하면 좋을지 고민이 많이 필요할 수도 있습니다. 이미 구현된 부분들을 잘 살피고 어떤 식으로 구현해야 할 지 스스로 채워주세요.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct queue {
5      int front;
6      int rear;
7      int size;
8      int capacity;
9      int* array;
10 } Queue;
11
12 Queue* createQueue(int capacity) {
13     Queue* queue = (Queue *)malloc(sizeof(Queue));
14     queue->capacity = capacity;
15     queue->front = 0;
16     queue->size = 0;
17     queue->rear = capacity-1; // 왜 이렇게 초기화 했는지 잘 생각해 보세요!
18 }

```



수강완료

```

21
22  int isFull(Queue* queue) {
23      return (queue->size == queue->capacity);
24  }
25
26  int isEmpty(Queue* queue) {
27      return (queue->size == 0);
28  }
29
30  void enqueue(Queue* queue, int item) {
31      if (isFull(queue)) {
32          return;
33      }
34      // 이 부분을 구현해 주세요!
35      printf("%d enqueued to queue\n", item);
36  }
37
38
39  int dequeue(Queue* queue) {
40      if (isEmpty(queue)) {
41          return -9999;
42      }
43      int item = 0;
44      // 이 부분을 구현해 주세요!
45      return item;
46  }
47
48
49  int main() {
50      Queue* queue = createQueue(1000);
51
52      enqueue(queue, 10);
53      enqueue(queue, 20);
54      enqueue(queue, 30);
55      enqueue(queue, 40);
56
57      printf("%d dequeued from queue\n\n", dequeue(queue));
58      return 0;
59  }

```

위의 Main 함수를 실행시키면 Queue 출력 결과가 아래와 같이 나와야 합니다.

```

10 enqueued to queue
20 enqueued to queue
30 enqueued to queue
40 enqueued to queue
10 dequeued from queue

```

```

Front item is 20
Rear item is 40

```

Queue 도 연결리스트를 이용해서 구현할 수도 있습니다. 과제와 상관없이 Queue를 연결리스트로도 구현해 보면 어떨까요?

3. 핵심 개념

#Queue #연결리스트 #Queue 구현

✓ 미션 4.

1. 미션 제목



수강완료

2. 지시문

이번에는 연결리스트의 응용 문제를 풀어보겠습니다. 연결리스트가 하나 주어졌을 때 해당 연결 리스트의 뒤에서 k번째 노드의 :
 었는지 알아낼 수 있을까요? 예를 들어, 9->8->4->14->5 라는 리스트가 주어질 때 뒤에서 2번째 노드를 출력하라고 하면 14가
 이 되어야 합니다.
 연결리스트가 이미 만들어져 있다고 가정하고, 아래와 같은 함수가 주어졌을 때 뒤에서 k번째 노드를 출력할 수 있을까요?

```

1  typedef struct node{
2      int data;
3      struct node* next;
4  } Node;
5
6  void append(Node* head, int data) {
7      // 이 부분을 작성해 주세요!
8  }
9
10 int getLastNode (Node* head, int k) {
11     // 이 부분을 작성해 주세요!
12 }
13
14 void printList(Node* head) {
15     // 이 부분을 작성해 주세요!
16 }
17
18
19 int main() {
20     Node* head = (Node*)malloc(sizeof(Node));
21     append(head, 9);
22     append(head, 8);
23     append(head, 4);
24     append(head, 14);
25     append(head, 5);
26
27     printList(head);
28
29     printf("%dth last node is %d\n", 2, getLastNode(head, 2));
30 }

```

main 함수에는 정렬된 연결 리스트 1개가 미리 만들어져 있다고 가정합니다. 여러분이 작성하신 함수 getLastNode 함수에 매
 에서 작성된 연결 리스트와 k를 파라미터로 주었을 때 뒤에서 k 번째 노드의 값이 나올 수 있을까요?

입력값:

미리 작성된 연결리스트와 k번째를 지칭하는 파라미터, 예를 들어 9->8->4->14->5 라는 리스트가 main 함수에 작성되어 있고
 함수에서 해당 리스트와 k=2로 getLastNode를 호출

출력값:

9 8 4 14 5

2th last node is 14

3. 핵심 개념

#연결리스트 #리스트 삽입 #Slow Pointer #Fast Pointer

*** 미션(4문제) 중에 회의를 통해 1개를 선택해서, 답을 목요일에 리드 부스터님이 팀별로 1개만 :
 하면 됩니다.**

*** 모든 미션을 푸는 것은 권장하나, 코치님의 피드백은 1개의 미션에 대해서만 제공됩니다!**

*** 6주차 미션같이 C에서 malloc function을 사용하는 경우 memset function을 이용하여 변수
 memory를 초기화하는 code를 추가하는 것을 강력 추천합니다. 예를 들어 아래와 같이 memset을
 출하면 됩니다.**



수강완료

💡 문제를 풀다가 도저히 풀 수 없을때는 어떻게 하나요?

🗨 팀 내 해결이 안되는 문제 혹은 구글링으로도 이해가 어려운 질문사항은 왼쪽의 "코치질문" 게시판에 질문해주세요!

여러분들의 담당 코치가 답변을 해줄 것이나 해당 질문에 답을 아는 동료부스터 누구나 답변을 남길 수 있으며 C 코칭스터디는 동료피드백을 통한 서로의 성장을 응원합니다. ♥

💡 우수 미션으로 뽑히고 싶은데 기준이 어떻게 되나요?

🗨 제출한 미션에 우수답변으로 문항 별 2~3개정도 선별하여 매주 금요일 라이브 특강에서 소개합니다.
우수미션 선정 기준은 다음과 같습니다.

1)코드의 정확성,효율성 2) 아이디어의 참신함, 창의성 3) 다양한 방법으로의 시도

💡 제출한 미션에 대한 피드백은 언제 주어나요?

🗨 제출한 미션에 대한 피드백은 다음날 금요일 정오 전까지 여러분의 담당 코치가 제공합니다.

또한 매주 금요일 8시 진행되는 라이브 특강을 통해 다른 팀의 미션 해결 사고방법과 다른 부스터들의 창의적인 소개할 예정입니다.

공유하기

좋아요 42

연관 토론 168

[9조] 6주차 - 팀 미션 이야기 나눠요.



9조리드로zeny 2021.02.20 0 2

[1조_6주차] 미션 이야기 나눠요



1조리드nauni 2021.02.20 0 1

[9조] 6주차 미션



9조리드myunghoonju 2021.02.20 0 2

[9조] 6주차 미션 얘기 나눠요~



9조리드으아아 2021.02.21 0 0

[7조] 6주차 학습인증 남겨주세요 :)



7조리드습니 2021.02.21 0 15



수강완료

[9조] 6주차 미션 얘기가네요~



9조dada 2021.02.22 0 7

[4조] 6주차 미션 인증!



4조리드난데없는개발자 2021.02.22 0 0

6주차 학습인증 공간입니다



10조리드쑥이 2021.02.22 0 11

[1조] 6주차 미션 얘기가네요 ~:)



1조리드lechae 2021.02.22 0 4

[3조] 6주차 미션 얘기가네요~



3조리드오잉 2021.02.22 0 4

[4조] 6주차 미션 얘기 나눠요.



4조리드BB 2021.02.22 0 2

[6조] 6주차 학습인증 게시판!



6조리드인프제 2021.02.22 0 17

[1조] 6주차 미션얘기 나눠요~



1조리드예절 2021.02.22 0 2

[10조] 6주차 학습 인증 남겨주세요~~~



10조리드시우 2021.02.22 0 11

[8조] 6주차 학습인증 부탁드립니다



8조리드너왜연락안받아 2021.02.22 0 6

[3조] 6주차 팀 미션 이야기



3조리드Harry 2021.02.22 0 4

[10조] 6주차 학습인증 남겨주세요~



10조리드rumpia 2021.02.22 0 14

[3조] 6주차 팀 미션 얘기가네요



3조리드Binsu 2021.02.22 0 2

[1조] 6주차 미션이야기 나눠요!



1조리드GoMok 2021.02.22 0 4



수강완료

