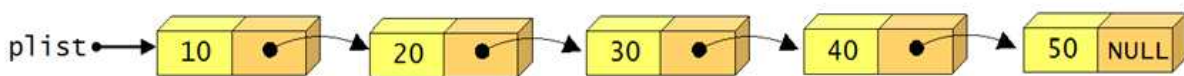


구조체의 멤버로는 포인터 변수가 들어갈 수 있다. 이러한 점을 이용하는 대표적인 예시로는 연결리스트가 있다.

연결리스트는 Node라는 구조체를 기본으로 한다. 각 Node는 데이터를 저장하는 공간과 다음 노드를 가리키는 공간으로 이루어진다.(그림 1 참조)

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

「그림 1」 Node 구조체



「그림 2」 연결 리스트, 각 상자는 Node를 의미

연결리스트는 첫 번째 노드를 가리키는 헤드 포인터가 필요하다.(그림 2에서 plist) 연결 리스트에 노드가 하나도 없으면 헤드 포인터 값은 NULL이다.

(그림 3을 보면 헤드 포인터인 plist가 전역 변수로 선언되어 있고, 초기 값은 NULL인 것을 알 수 있다.)

이제 `Push(int data)` 함수를 만들어보자. Push 함수는 연결리스트의 제일 뒤에 새로운 노드를 추가한다.

Push 함수에서는 지역변수 두 개가 필요하고, 둘 다 Node 포인터 타입이다. 하나는 새로운 노드를 할당받을 변수이고, 하나는 현재 마지막 노드를 찾기 위한 변수이다.

Push 함수의 알고리즘은 다음과 같다.

1. 새로운 노드를 할당받는다.(동적 할당)
2. 제일 뒤에 추가하므로, 새로 추가하는 노드의 다음은 NULL이다.
3. 매개변수로 받은 data를 노드에 저장한다.
4. 연결리스트가 비어있으면(헤드 포인터가 NULL이면) 새로 할당받은 노드가 시작 노드가 되므로 헤드 포인터가 시작 노드를 가리키게 한다.

5-1. 연결리스트가 비어있지 않으면, 현재 제일 뒤에 있는 노드를 찾아야 한다. 이는 주어진 Print 함수를 참고하라. Print 함수의 while 문에서 조건을 조금만 수정하면 제일 뒤에 있는 노드를 찾을 수 있다.

5-2. 그리고 제일 뒤에 있는 노드의 next가 새로 할당받은 노드를 가리키게 한다.

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  typedef struct node
5  {
6      int data;
7      struct node* next;
8  }Node;
9
10 Node* plist = NULL;
11
12 void Push(int data);
13 void Print();
```

「그림 3」

```
15 int main(void)
16 {
17     int menu, input;
18
19     while (1)
20     {
21         printf("할 작업을 선택하시오(1 : push, 2 : print, else : exit) : ");
22         scanf("%d", &menu);
23
24         if (menu == 1)
25         {
26             printf("데이터 입력 : ");
27             scanf("%d", &input);
28             Push(input);
29         }
30         else if (menu == 2)
31             Print();
32         else
33             break;
34     }
35 }
```

「그림 4」

```

void Print()
{
    Node* tmp = plist;

    if (plist == NULL)
    {
        printf("리스트가 비어있습니다.\n");
    }
    else
    {
        while (tmp != NULL)
        {
            printf("%d\t", tmp->data);
            tmp = tmp->next;
        }
        printf("\n");
    }
}

```

「그림 5」

```

C:\WINDOWS\system32\cmd.exe
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 1
데이터 입력 : 3
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 1
데이터 입력 : 5
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 1
데이터 입력 : 6
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 2
3 5 6
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 1
데이터 입력 : 8
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 1
데이터 입력 : 10
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 2
3 5 6 8 10
할 작업을 선택하시오(1 : push, 2 : print, else : exit) : 3
계속하려면 아무 키나 누르십시오 . . .

```

「그림 6」