

# STL과 즐거운 프로그래밍



알아야 하는 것  
C언어

알면 좋은 것  
자료구조



```
#include <stdlib.h>
```

**stdlib**

standard library

standard template library

**template???**

두 값을 비교해서 큰  
값을 반환하는 함수를  
만들어 보자.

```
int Max(int a,int b)
{
    return a > b ? a : b;
}
```

**float은?**

```
float Max(float a, float b)
{
    return a > b ? a : b;
}
```

미안한데...  
long long도

```
long long Max(long long a,  
              long long b)  
{  
    return a > b ? a : b;  
}
```

이건 아닌거 같다.

**한번에 만들고 싶다.**

```
template <typename T> T  
Max(T a, T b)  
{  
    return a>b ? a : b;  
}
```

# 참고로

- 요건 함수템플릿이고 클래스템플릿도 있어요
- 이런 프로그래밍 스타일을 제네릭 프로그래밍이라고 불러요

standard template library

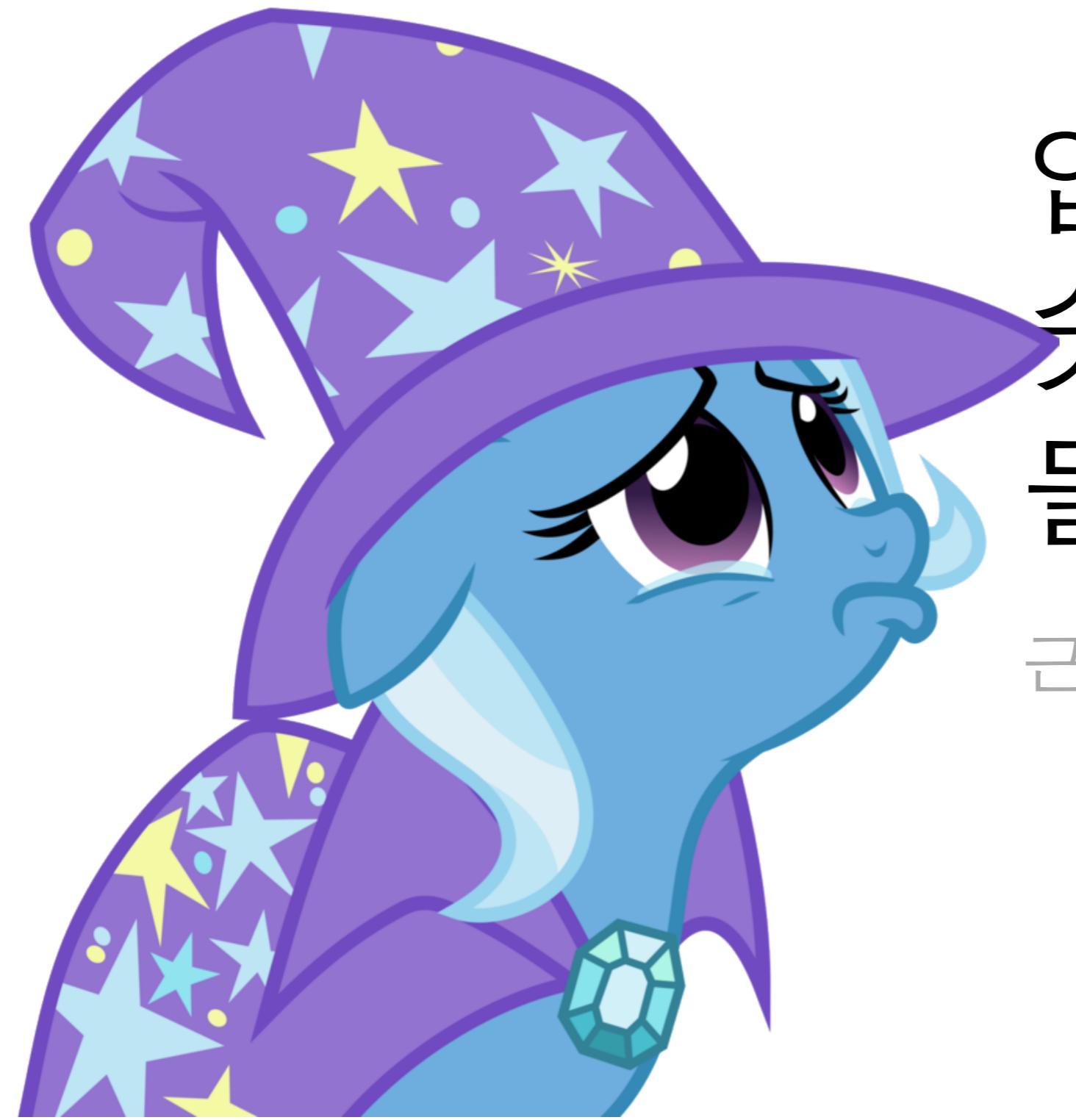
## C++ 의 표준 라이브러리

컨테이너(어떻게 저장하는지 생각 안해도  
빠르게 알아서 저장)와  
알고리즘(검색도 해주고 정렬도 해줌)

# 근데 잠깐 C++이요?

- ACM ICPC
- 고성능 프로그램
- 멀티플랫폼 지원
- 그리고 과제, 게임 만들 때는 C++
- 지금 당장 C++에 관심이 없어도 C로 과제 할 때 C++의 기능을 빌려보자.

**그래서 STL**



입력으로  
숫자가 계속 계속  
들어옵니다.

근데 크기는 얼마인지 잘...



```
#include<stdio.h>
```

```
#include<stdio.h>
```

```
#include<stdio.h>
```

```
int main()
{
```

```
#include<stdio.h>
```

```
int main()
{
```

```
    int a[...];
```

```
#include<stdio.h>

int main()
{
    int a[...];
}
```

```
#include<stdio.h>

int main()
{
    int a[...];
}
```







```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int *ar, size;
    size = 5;
    ar = (int*)malloc(size*sizeof(int));

    //앗! 배열이 모자라다
    size = 10;
    ar = (int*)realloc(ar, size*sizeof(int));
    free(ar);
}
```



간단하게 말해서  
이 문제를 해결합시다.

Vector= 벡터



```
#include<vector>
```

```
#include<vector>

using namespace std;
int main()
{
```

```
#include<vector>

using namespace std;
int main()
{
    vector<int> vector1;
```

```
#include<vector>

using namespace std;
int main()
{
    vector<int> vector1;
    vector1.push_back(10);
    vector1.push_back(20);
    vector1.push_back(30);
```

```
#include<vector>

using namespace std;
int main()
{
    vector<int> vector1;
    vector1.push_back(10);
    vector1.push_back(20);
    vector1.push_back(30);
    int Count = vector1.size();
```

```
#include<vector>

using namespace std;
int main()
{
    vector<int> vector1;
    vector1.push_back(10);
    vector1.push_back(20);
    vector1.push_back(30);

    int Count = vector1.size();

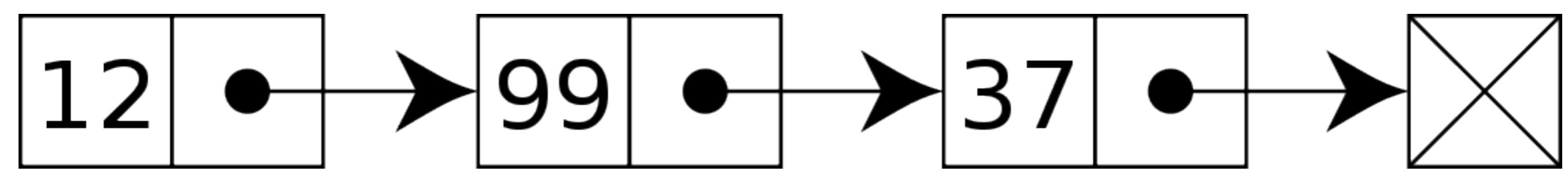
    for(int i=0;i<Count;++i)
        cout << vector1[i] << endl;
    return 0;
}
```

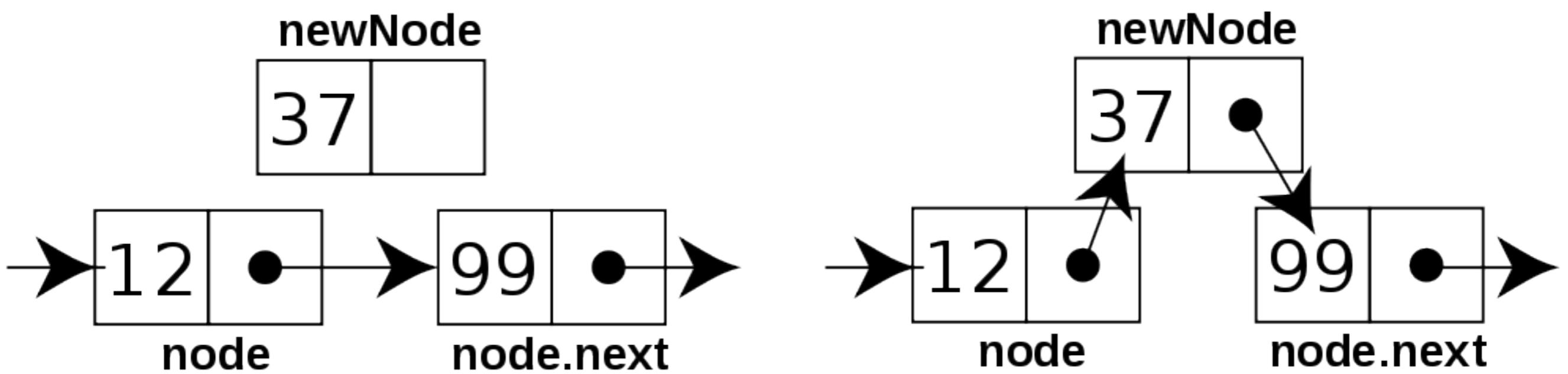
# Vector의 문제점

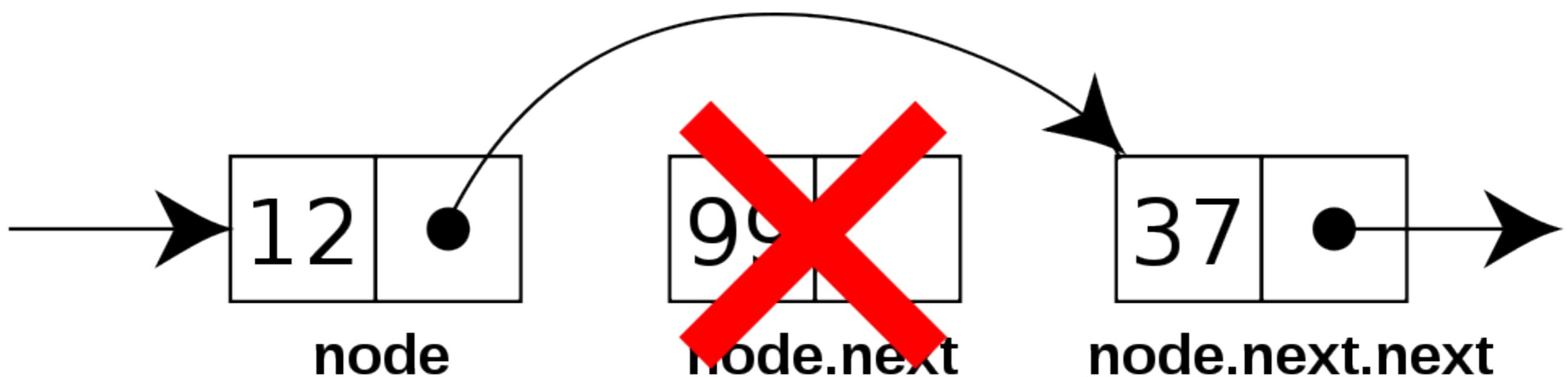
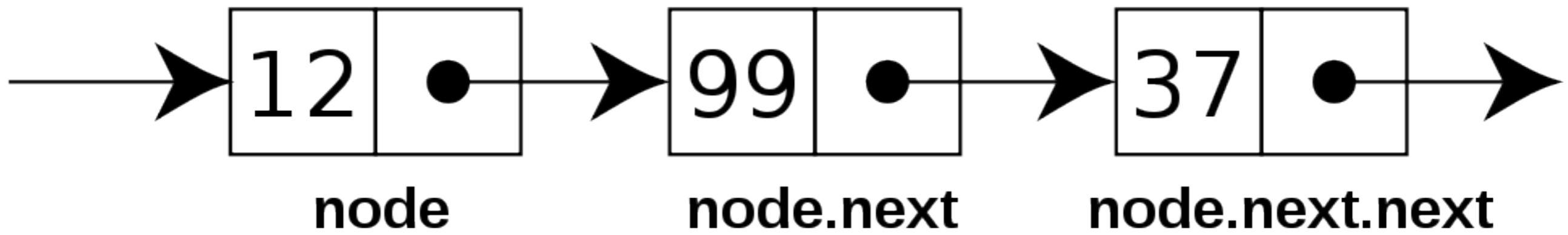
- Vector는 내부적으로 배열로 만들어져 있다.
- 배열에 중간에 껴 넣으려면 뒤로 밀어야 한다.

나는 중간에 껴 넣고  
싶다.

# 연결리스트







아...

혐오스러운 포인터

```
#include<list>

list<int> L;

L.insert(L.begin() + 3, 100);
```

# 그 외

- deque
- stack
- queue
- priority\_queue



지금까지는  
시퀀스 컨테이너

호야 친구들이  
중간고사를 보았습니다.

10점



곰팅이

70점



햄순이

100점



호야

```
#include <map>
```

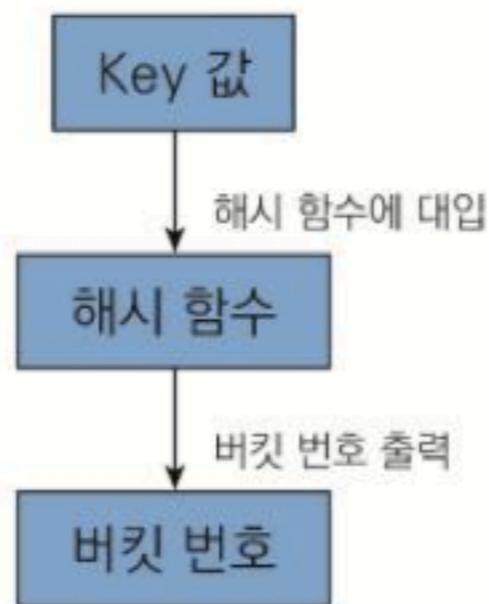
```
map<string,int> score;
```

```
score[“곰팅이”]=10;
```

```
score[“햄순이”]=70;
```

```
score[“호야”]=100;
```

# hash\_map



버킷의 빈 슬롯에 넣는다

버킷 번호	슬롯 1	슬롯 2	슬롯 3
0			
1			
...			
N			

해시 테이블

# map VS. hash\_map

map	hash_map
정렬해서 저장	정렬하지 않음
빠른 검색	완전 빠른 검색

# 연관 컨테이너

- map, set, hash\_map, hash\_set
- multi\_map, multi\_set, hash\_multimap, hash\_multiset

# 알고리즘

#include < algorithm >

find

find\_if

for\_each

reverse

random\_shuffle

sort

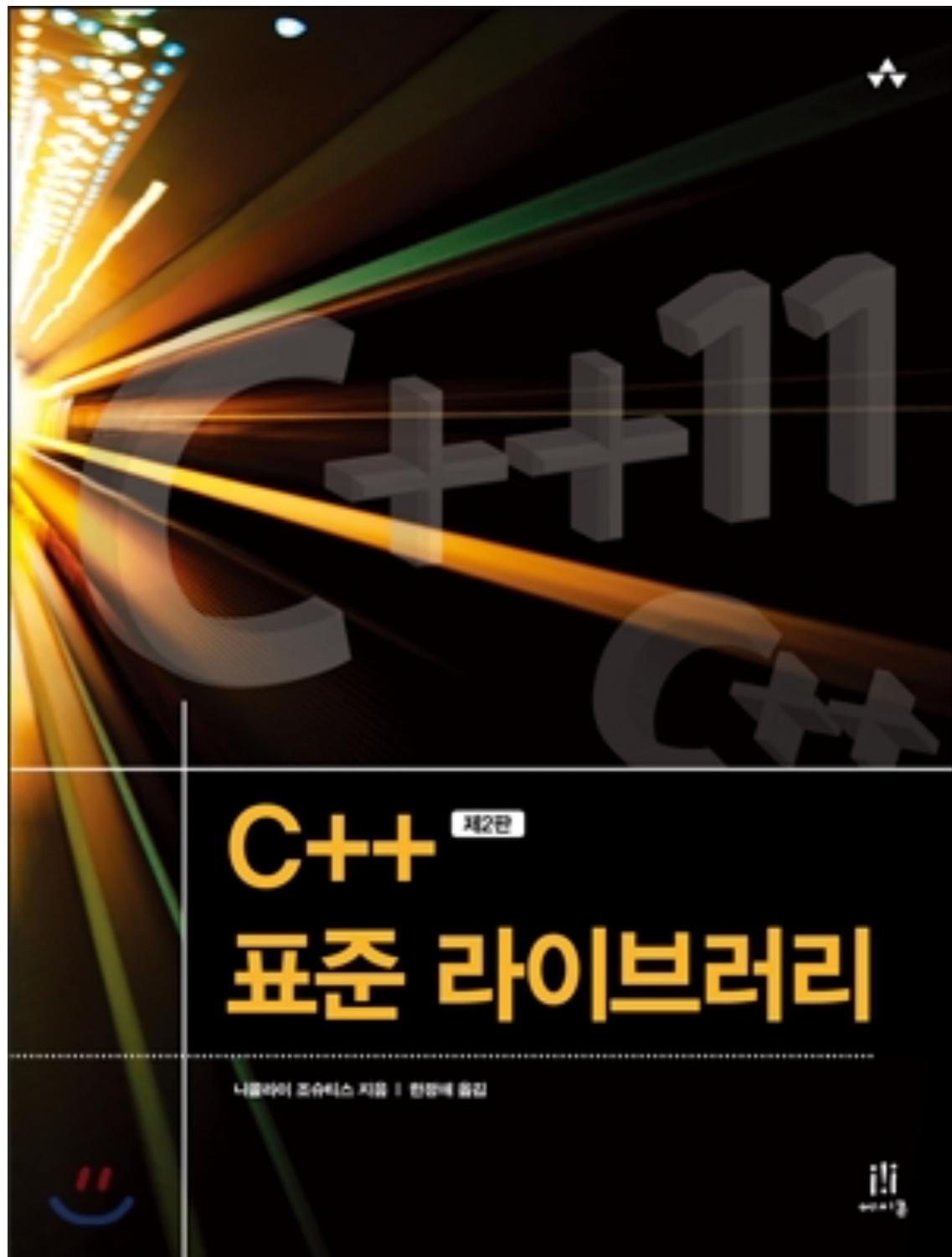
```
vector<int> vec1(10);  
generate( vec1.begin(),  
          vec1.end(),  
          rand );  
  
sort( vec1.begin(),  
      vec1.end() );
```

**더 알고 싶어요!!!**

그럴리가 없지만



- 한빛미디어 eBook
- 무료
- [http://  
www.hanb.co.kr/  
ebook/look.html?  
isbn=9788979149937](http://www.hanb.co.kr/ebook/look.html?isbn=9788979149937)



- 1200 페이지
- 책 아닌 흉기
- 혼자 이걸 어떻게 번역했을까?
- C++ 11 개정판

# C++||

- 2011년에 C++11이라고 부르는 C++ 표준이 나왔다.
- C++ 0x라고 불리던...

# auto

- `list<int>::iterator iter=myList.begin();`
- `auto iter=myList.begin();`

# lambda

```
auto Iter = find_if( Users.begin(), Users.end(),
[](User& tUser) -> bool { return true ==
tUser.IsDie();});
```

```
struct FindDieUser
{
    bool operator()(User& tUser) const
    {
        return tUser.IsDie();
    }
};
```

# range base for

```
int myList[] = {1,2,3,4,5};
```

```
for( auto i : myList )  
{  
    printf("%d",i);  
}
```

# range base for

```
vector<int> myList;  
myList.push_back(1);  
myList.push_back(2);  
myList.push_back(3);  
for( auto i : myList )  
{  
    printf("%d",i);  
}
```

그래도 C++의 기능이  
부족하다면

<http://www.boost.org/>

# 결론

- 필요한 자료구조를 직접 만들지 않아도 된다. 시간을 절약하자.
- 어차피 만들어도 버그 투성이. 너의 코드를 믿는 너를 믿지 않는다.
- 표준 라이브러리이므로 사용 방법이 언제나 같아서 누구나 소스코드를 이해

발표자료

[github.com/hoyajigi>HelloSTL](https://github.com/hoyajigi>HelloSTL)

