



세종대학교

---

# Report

명품 C++ 프로그래밍 5장 실습문제 6, 8

제출일 2018.04.14

전공 디지털콘텐츠학과

학번 16013093

이름 박상우

## 문제 6

사이즈를 인자로 받아 그 사이즈 만큼 동적할당하는 생성자를 만들었습니다.

스택은 0부터 쌓이기 때문에 tos를 0으로 설정했고 size를 멤버변수 size에 저장했습니다.

```
MyIntStack::MyIntStack(int size){
    tos = 0;
    p = new int[size];
    this->size = size;
}
```

push 함수는 tos가 size와 같으면 스택이 꽉 찬 것이므로 false를 리턴하고 자리가 비어있으면 n을 저장하게 했습니다.

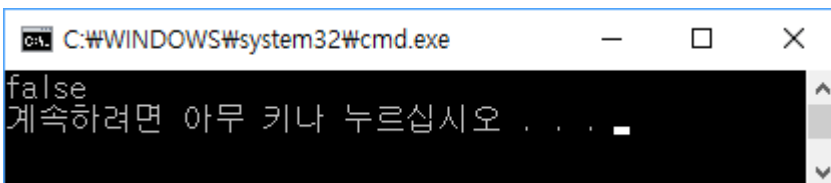
```
if(tos == size)
    return false;
else{
    p[tos] = n;
    tos++;
    return true;
}
```

pop 함수는 tos가 0이면 스택이 비어있는 것이므로 false를 리턴하고 아니라면 참조를 이용해 n에 데이터를 저장하게 했습니다.

```
if(tos == 0)
    return false;
else{
    tos--;
    n = p[tos];
    return true;
}
```

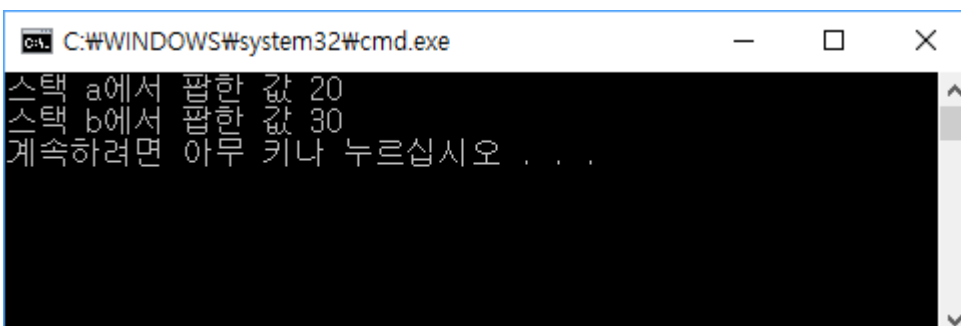
복사의 경우 깊은 복사가 될 수 있게 새로 동적할당하고 반복문으로 데이터까지 복사하였습니다.

```
MyIntStack a(2);
a.push(10);
a.push(20);
cout << boolalpha << a.push(30);
cout << endl;
// main()을 수정해 사이즈를 2로 설정하고 3번째 push할 때 false가 리턴됨을 확인했습니다.
```



```
C:\WINDOWS\system32\cmd.exe
false
계속하려면 아무 키나 누르십시오 . . .
```

결과입니다.



```
C:\WINDOWS\system32\cmd.exe
스택 a에서 팝한 값 20
스택 b에서 팝한 값 30
계속하려면 아무 키나 누르십시오 . . .
```

## 문제 8

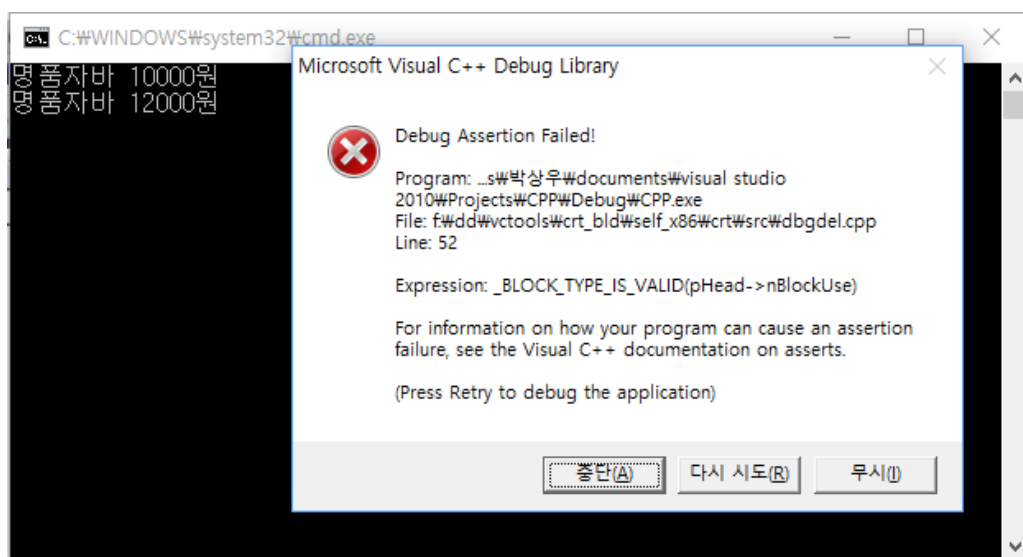
(1) 생성자는 문자열 복사를 위해 `strlen()`으로 크기를 구한 뒤 크기+1 만큼 동적할당 하고 title을 복사하게 했습니다.

```
Book::Book(const char * title, int price){
    int size = strlen(title);
    this->title = new char[size+1];
    strcpy(this->title, title);
    this->price = price;
}
```

set 함수는 문자열과 가격을 멤버변수에 복사하게 구현했습니다.

```
void Book::set(char*title, int price){
    strcpy(this->title, title);
    this->price = price;
}
```

(2) 디폴트 복사생성자는 title 포인터를 복사하기 때문에 두 포인터가 한 주소를 나타내게 됩니다.



“명품 C++”과 “명품자바”가 같은 주소를 나타내기 때문에 `java.set()`에서 “명품 C++” 데이터가 바뀌고 명품자바를 출력하는 모습입니다. 또한 두번째 소멸자가 호출될 때 이미 첫번째 소멸자에서 반환했고 잘못된 메모리 접근으로 오류가 나게 됩니다.

(3) 깊은 복사를 위해 복사 생성자를 추가했습니다.

```
Book::Book(Book &a){
    int size = strlen(a.title);
    this->title = new char[size+1];
    strcpy(this->title, a.title);
    this->price = price;
}
```

새롭게 동적할당하고 복사했기 때문에 정상적으로 복사가 일어나게 됩니다.

