

Week2

- misc -

안태진(taejin7824@gmail.com)

GitHub(github.com/taejin1221)

상명대학교 소프트웨어학과

201821002

Contents

- Notice
- 문제 풀이
- 문제
- Dynamic Allocation

Notice

- GitHub
 - GitHub 링크
 - <https://github.com/Tutoring-Taejin>
- Slack
 - 초대 링크
 - https://join.slack.com/t/tutoring-taejin/shared_invite/zt-nz1m4nwl-A3St1yfei2iKKCokIArLMA

Contents

- Notice
- 문제 풀이
- 문제
- Dynamic Allocation

Contents

- Notice
- 문제 풀이
- 문제
- Dynamic Allocation

문제

- 과제
 - LeetCode 01 : Two Sum
 - <https://leetcode.com/problems/two-sum/>
 - Baekjoon 2920 : 음계
 - <https://acmicpc.net/problem/2920>
- 과제 제출
 - 깃허브에 업로드! 업로드 안하면 인정 x

Contents

- Notice
- 문제 풀이
- 문제
- Dynamic Allocation

Dynamic Allocation

- Dynamic Allocation
 - Runtime에 메모리를 할당 받는 방법

- Static vs Dynamic

분야	Static	Dynamic
할당 위치	Stack	Heap
크기 결정 시기	Compile time	Runtime
할당 방향	High -> Low	Low -> High
사용자가 직접 관리?	X	O

Dynamic Allocation

- C언어
 - 할당
 - stdlib header file의 malloc()
 - 해제
 - stdlib header file의 free()
- C++
 - 할당
 - new
 - 해제
 - delete

Dynamic Allocation

- Practice 1

- Dynamic Allocation

- Type* variable_name = new Type;

```
char* char1 = new char;  
short* short1 = new short;  
int* int1 = new int;  
long long* long1 = new long long;
```

- Size of Variables

- 단순 Pointer는 주소를 담고 있기 때문

```
cout << sizeof( char1 ) << endl;  
cout << sizeof( short1 ) << endl;  
cout << sizeof( int1 ) << endl;  
cout << sizeof( long1 ) << endl;
```

```
8  
8  
8  
8  
=====
```

- Size of allocated memory

- 각 type의 크기 할당

```
cout << sizeof( *char1 ) << endl;  
cout << sizeof( *short1 ) << endl;  
cout << sizeof( *int1 ) << endl;  
cout << sizeof( *long1 ) << endl;
```

```
1  
2  
4  
8  
=====
```

Dynamic Allocation

- Practice 1

- Pointer 사용과 같음
- 할당된 메모리는 꼭! 삭제
 - delete keyword

```
delete char1;  
delete short1;  
delete int1;  
delete long1;
```

```
*int1 = 10;  
cout << int1 << endl;  
cout << *int1 << endl;  
  
cout << "===== " << endl;  
  
*int1 = 20;  
cout << int1 << endl;  
cout << *int1 << endl;
```

```
0x7fe25b405920  
10  
=====  
0x7fe25b405920  
20
```

Dynamic Allocation

- Practice 2

- 여러 값 한번에 할당 (Array)

- `Type* arr = new Type[num];`
 - `Type* arr = new Type[num] { e1, e2, ... }`

```
int* nums_heap1 = new int[5] { 1, 2, 3, 4, 5 };  
int* nums_heap2 = new int[5] { 6, 7, 8, 9, 10 };
```

- 여러 값 한번에 해제

- `delete[] arr;`

```
delete[] nums_heap1;  
delete[] nums_heap2;
```

Reference

- 태진's 자료들
- 태진's Brain
- [wikipedia](#)
- [소년코딩](#)

감사합니다!
