메모리의 동적 할당 dynamic allocation

2023 국민대학교 소프트웨어학부

포인터와 배열: 배열의 이름은 포인터로 취급



```
#include <iostream>
using namespace std;
                                                                                               &a[0] = a
                                                      1245008
int main()
                                                      1245009
                                                                                       a[0]
                                                      1245010
    int a[] = \{ 10, 20, 30, 40, 50 \};
                                                      1245011
                                                                                              &a[1]
                                                      1245012
    cout << "&a[0] = " << &a[0] << endl;
                                                      1245013
                                                                                       a[1]
    cout << "&a[1] = " << &a[1] << endl;
                                                      1245014
    cout << "&a[2] = " << &a[2] << endl;
                                                      1245015
                                                      1245016
                                                                                              &a[2]
    cout << "a = " << a << endl;
                                                      1245017
                                                                                       a[2]
                                                      1245018
    return 0;
                                                      1245019
                                                      1245020
                                                                                              ·&a[3]
                                                                                       a[3]
                                                      1245021
                                                      1245022
&a[0] = 1245008
                                                      1245023
&a[1] = 1245012
&a[2] = 1245016
                                                                   메모리
a = 1245008
```



포인터 연산



```
#include <iostream>
                                                                                               *(a+3)
                                                                      *(a+1)
using namespace std;
int main()
    int a[] = { 10, 20, 30, 40, 50 };
                                        변소값
                                                                a[0]
                                                                                        a[2]
                                                                                                     a[3]
                                       변수이름
                                                                            a[1]
    cout << "a = " << a << endl;
    cout << "a + 1 = " << a+1 << endl;
    cout << "*a = " << *a << endl;
    cout << "*(a+1) " << *(a+1) << endl;
    return 0;
```



```
a = 1245008
a + 1 = 1245012
*a = 10
*(a+1) = 20
```

```
#include <iostream>
    using namespace std;
    int main(){
      int iarr[20], *ip = iarr;
     char carr[20] = "characters", *cp = carr;
     double darr[20], *dp = darr;
      cout << "ip : " << ip << endl;
      cout << "ip+1 : " << ip+1<< endl;
10
11
      cout << "cp : " << cp << endl;
12
      cout << "cp+1 : " << cp+1<< endl;
      cout << "(void *)cp : " << (void *)cp << endl;
13
      cout << "(void *)(cp+1) : " << (void *)(cp+1)<< endl; dp+1 : 0x7ffe342a1fe8
14
      cout << "dp : " << dp << endl;
15
      cout << "dp+1 : " << dp+1<< endl;
16
17 }
```

```
ip : 0x7ffe342a1f90
ip+1 : 0x7ffe342a1f94
cp : characters
cp+1 : haracters
(void *)cp : 0x7ffe342a2080
(void *)(cp+1) : 0x7ffe342a2081
dp : 0x7ffe342a1fe0
```

포인터를 사용한 방법의 장점

- 인덱스 표기법보다 빠르다.
 - 원소의 주소를 계산할 필요가 없다.

```
int get_sum1(int a[], int n)
{
    int i;
    int sum = 0;

    for(i = 0; i < n; i++)
        sum += a[i];
    return sum;
}</pre>
```

```
int get_sum2(int a[], int n)
{
    int i;
    int *p;
    int sum = 0;

    p = a;
    for(i = 0; i < n; i++ )
        sum += *p++;
    return sum;
}</pre>
```

인덱스 표기법 사용



포인터 사용



동적 할당 메모리의 개념

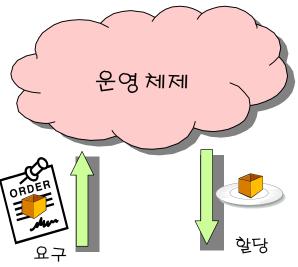
- 프로그램이 메모리를 할당받는 방법
 - 정적(static)
 - 동적(dynamic)
- 정적 메모리 할당
 - 프로그램이 시작되기 전에 미리 정해진 크기의 메모리를 할당받는 것
 - 메모리의 크기는 프로그램이 시작하기 전에 결정

```
int i, j;
int buffer[80];
char name[] = "data structure";
```

- 처음에 결정된 크기보다 더 큰 입력이 들어온다면 처리하지 못함
- 더 작은 입력이 들어온다면 남은 메모리 공간은 낭비

동적 메모리 할당

- 실행 도중에 동적으로 메모리를 할당받는 것
- 사용이 끝나면 시스템에 메모리를 반납
- 필요한 만큼만 할당을 받고 메모리를 효율적으로 사용
- new와 delete 키워드 사용



```
#include <iostream>
using namespace std;

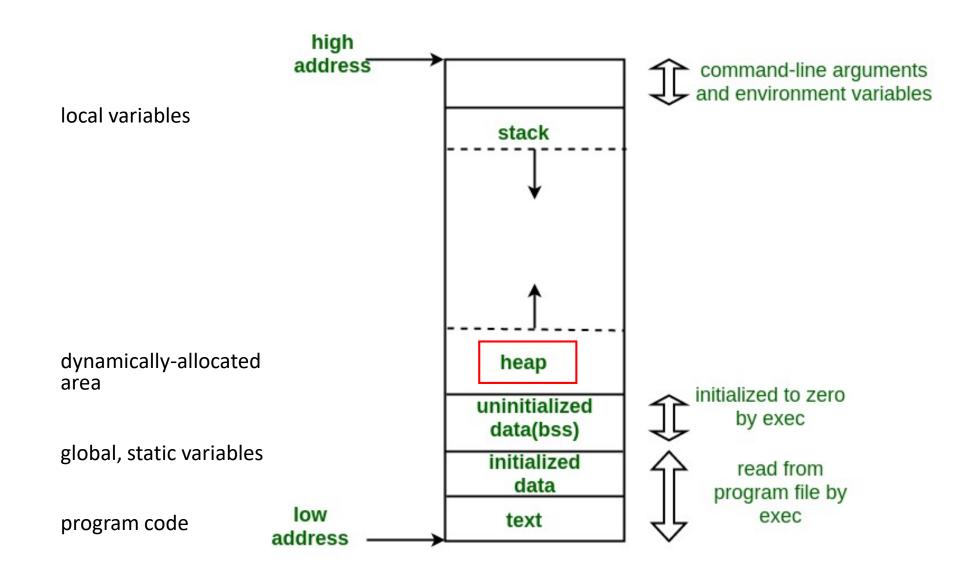
int main()
{
  int *p;
  p = new int;
  ...
}
```

프로그램

동적 메모리 할당 방법

- in C++ new / delete
- in C malloc() / free()

Typical Memory Layout of a C program

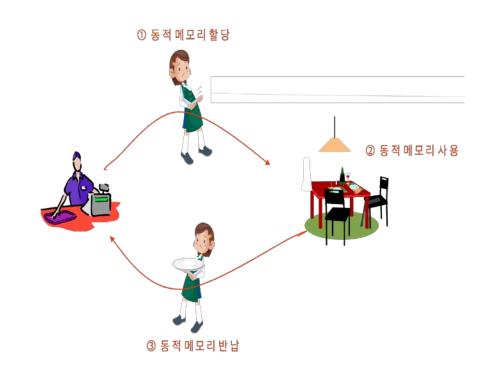


```
#include <stdio.h>
 1
       int globalVar;
 4
     □int main() {
 5
 6
           static int staticLocalVar;
 7
           int autoLocalVar;
 8
          printf("globalVar : %d\n", globalVar);
           printf("staticLocalVar: %d\n", staticLocalVar);
10
          printf("autoLocalVar : %d\n", autoLocalVar); // 컴파일러 오류
11
12
```

```
globalVar : 0
staticLocalVar: 0
```

동적 메모리 할당의 과정

```
#include <iostream>
using namespace std;
int main()
        int *pi; // 동적 메모리를 가리키는 포인터
                  // ① 동적 메모리 할당
        pi = new int;
                       // ② 동적 메모리 사용
        *pi = 100;
                       // ③ 동적 메모리 반납
        delete pi;
                     // 반납된 메모리를 재사용할 수 있음
        int *q = new int;
        return 0;
```



// pi == NULL 이면 delete 는 아무 것도 하지 않는다. error 가 아님

```
#include <iostream>
    using namespace std;
    int main(){
      int *p, *q;
    p = new int;
    cout << "p = " << p << endl;
     q = new int;
      cout << "q = " << q << endl;
10
11
      return 0;
12 }
```

```
1 #include <iostream>
   using namespace std;
    int main(){
    int *p, *q;
      p = new int;
      cout << "p = " << p << endl;
      delete p;
      q = new int;
      cout << "q = " << q << endl;
10
      return 0;
11
12 }
```

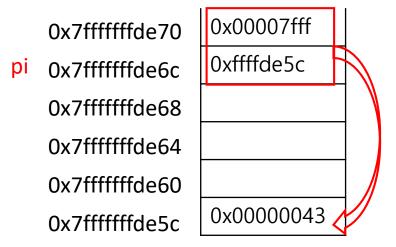
```
p = 0x562697778e70
q = 0x5626977792a0
```

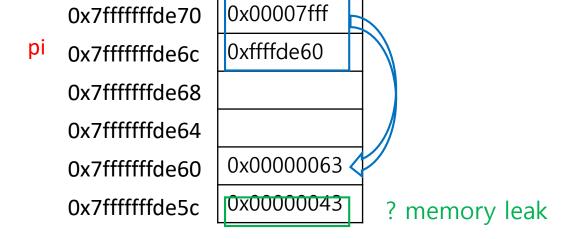
```
p = 0x5561ce586e70
q = 0x5561ce586e70
```

메모리 누수의 예제

```
void sub()
{
    int *p = new int; // ①
    *p = 0x43;

    p = new int; // ②
    *p = 0x63;
}
```



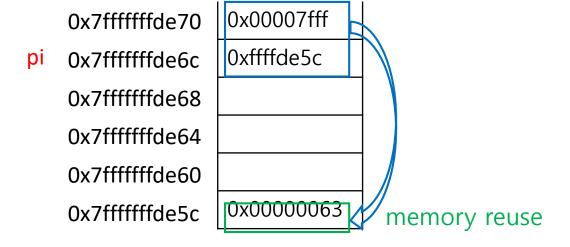


메모리 누수의 예제

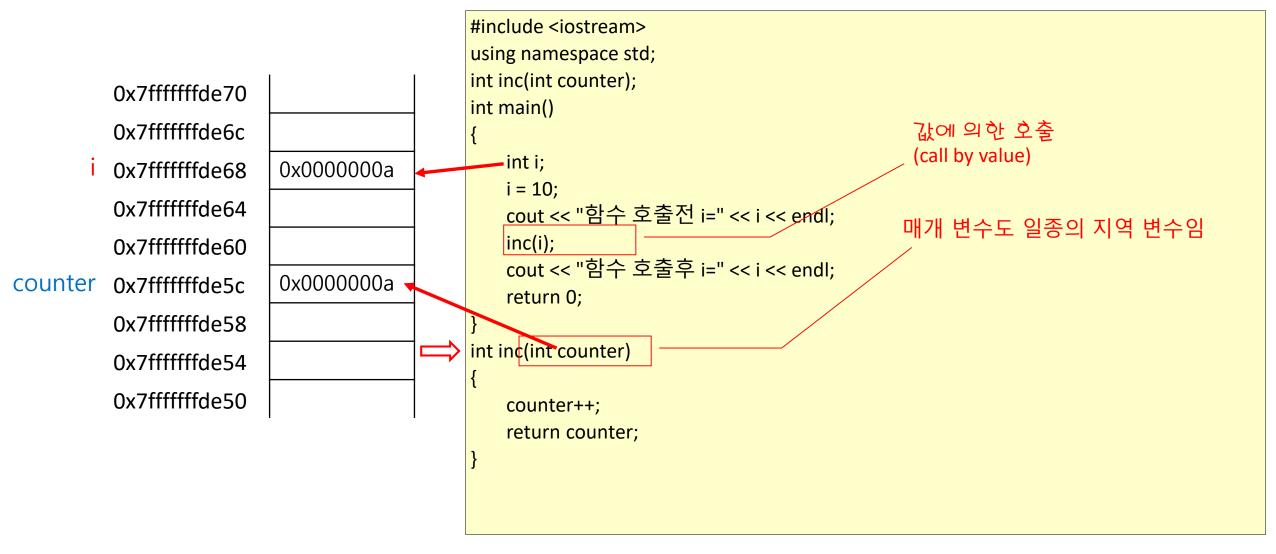
0x7fffffffde700x00007fff0x7fffffffde6c0xffffde5c0x7fffffffde680x7fffffffde640x7fffffffde600x7fffffffde5c0x7fffffffde5c0x00000043

```
void sub()
{
    int *p = new int; // ①
    *p = 0x43;
    delete p;

    p = new int; // ②
    *p = 0x63;
    delete p;
}
```

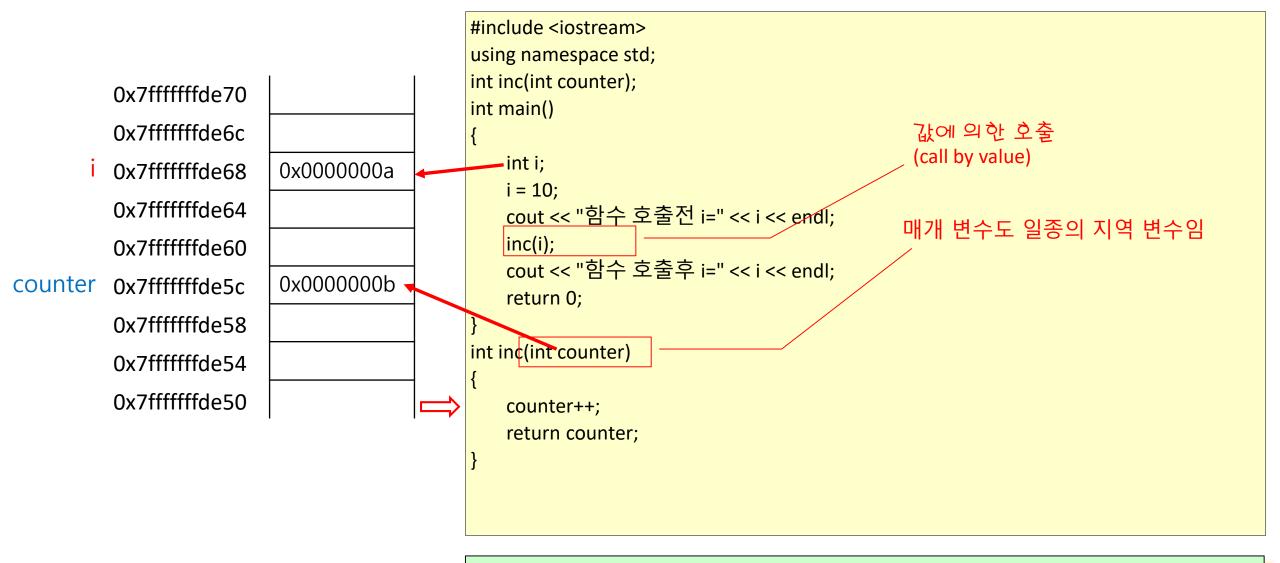


C 언어에서 함수의 인자 전달 방법은 call by value



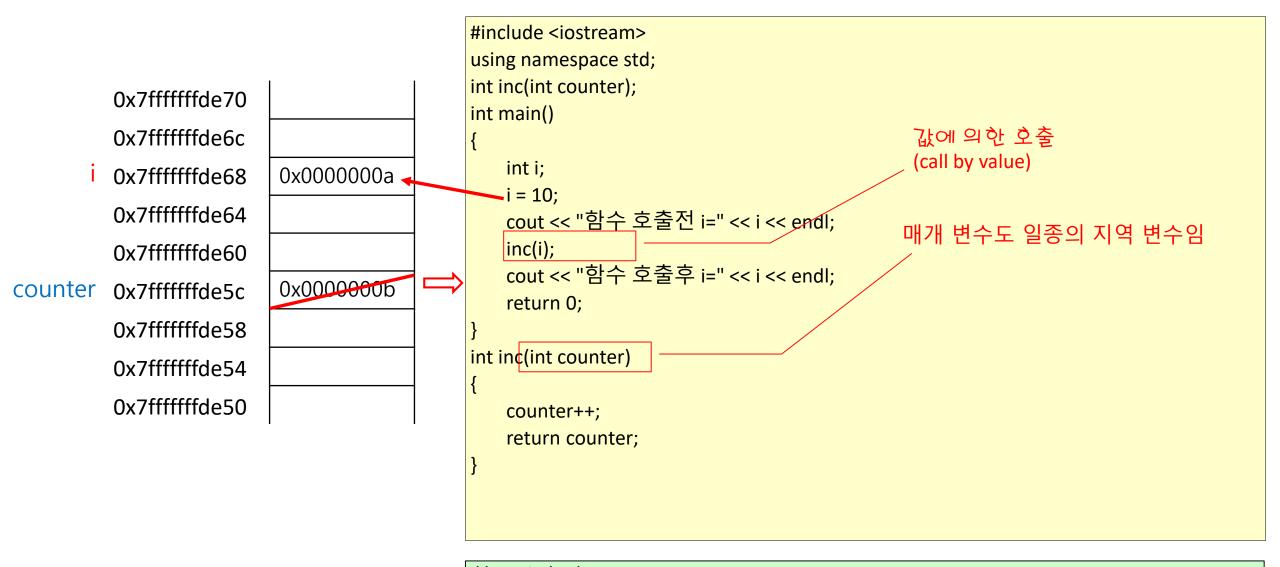
함수 호출전 i=10

C 언어에서 함수의 인자 전달 방법은 call by value



함수 호출전 i=10

C 언어에서 함수의 인자 전달 방법은 call by value



```
함수 호출전 i=10
함수 호출후 i=10
```

함수 인자로 배열 전달하기

• 실행문에서 배열의 이름만 쓰면 배열의 시작 주소를 의미한다.

→ 4	<pre>void f(int *p){ // f(int p[])</pre>
5	p[0] = 10;
6	p[1] = 20;
7	p[2] = 30;
8	}
9	
10	<pre>int main(){</pre>
11	int a[3] = {0};
12	
13	cout $<<$ (a == $\&a[0]$) $<<$ endl;
14	f(a); // same as f(&a[0]);
15	cout << "a[0] : " << a[0] << endl;
16	cout << "a[1] : " << a[1] << endl;
17	cout << "a[2] : " << a[2] << endl;
18	}

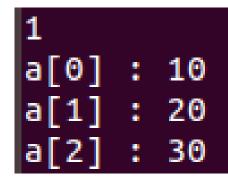
	0x7fffffffde70	0x00000000	
	0x7fffffffde6c	0x00000000	
a	0x7ffffffde68	0x00000000	
	0x7fffffffde64	0x00007fff	
p	0x7fffffffde60	0xffffde68	
	0x7fffffffde5c		
	0x7fffffffde58		
	0x7fffffffde54		
	0x7fffffffde50		

```
void f(int *p){ // f(int p[])
       p[0] = 10;
      p[1] = 20;
      p[2] = 30;
⇒ 8 }
   9
  10
      int main(){
        int a[3] = \{0\};
  11
  12
  13
        cout << (a == &a[0]) << endl;
        f(a); // same as f(\&a[0]);
  14
  15
        cout << "a[0] : " << a[0] << endl;
        cout << "a[1] : " << a[1] << endl;
  16
        cout << "a[2] : " << a[2] << endl;
  17
  18 }
```

	0x7fffffffde70	0x0000001e	
	0x7fffffffde6c	0x00000014	
a	0x7ffffffde68	0x0000000a	p[0]
	0x7fffffffde64	0x00007fff	
p	0x7fffffffde60	0xffffde68	
	0x7fffffffde5c		
	0x7fffffffde58		
	0x7fffffffde54		
	0x7fffffffde50		

```
void f(int *p){ // f(int p[])
     p[0] = 10;
      p[1] = 20;
      p[2] = 30;
 8
 9
10
    int main(){
      int a[3] = \{0\};
11
12
13
      cout << (a == &a[0]) << endl;
14
      f(a); // same as f(\&a[0]);
      cout << "a[0] : " << a[0] << endl;
      cout << "a[1] : " << a[1] << endl;
16
      cout << "a[2] : " << a[2] << endl;
18 }
```

0x7fffffffde70	0x0000001e
0x7fffffffde6c	0x00000014
0x7fffffffde68	0x0000000a
0x7fffffffde64	0x00007fff
0x7fffffffde60	0xffffde68
0x7fffffffde5c	
0x7fffffffde58	
0x7fffffffde54	
0x7fffffffde50	
	0x7fffffffde6c 0x7fffffffde68 0x7fffffffde64 0x7fffffffde60 0x7fffffffde5c 0x7fffffffde58 0x7fffffffde54



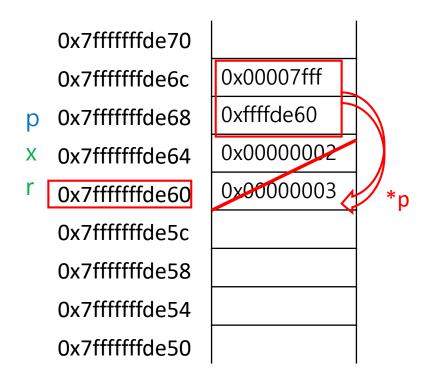
주의!!!

```
int *f(int x){
                                     지역 변수 r는 함수가 종료되면
                                     소멸되므로 그 주소를 반환하
      int r;
                                     면 안된다.!!
      r = x+1;
      return &r;
10
    int main(){
12
      int *p;
13
     p = f(2);
14
      cout << *p << endl;
15
16
```

	0x7fffffffde70	
	0x7ffffffde6c	0x00000000
р	0x7ffffffde68	0x00000000
X	0x7ffffffde64	0x00000002
r	0x7ffffffde60	0x00000003
	0x7ffffffde5c	
	0x7ffffffde58	
	0x7ffffffde54	
	0x7ffffffde50	

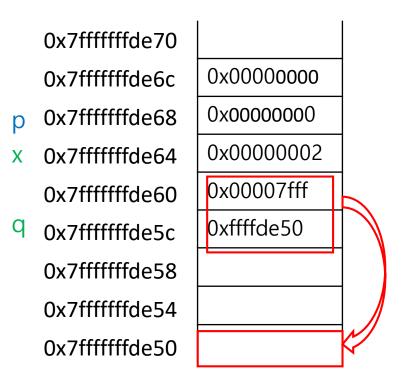
지역 변수의 주소를 반환하면, 함수가 종료되면 사라지기 때문에 오류

```
int *f(int x){
           int r;
           r = x+1;
            return &r;
     9
    10
         int main(){
    12
           int *p;
    13
           p = f(2);
    14
           cout << *p << endl;
\implies 15
    16
```



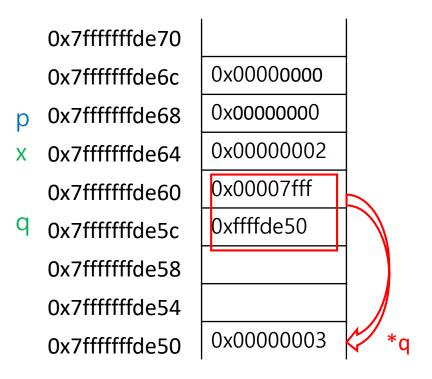
그러나 동적 할당받은 주소를 반환하는 것은 가능하다.

```
int *f(int x){
     int *q = new int;
      *q = x+1;
      return q;
 9
10
    int main(){
12
      int *p;
13
   p = f(2);
14
   cout << *p << endl;
16
```



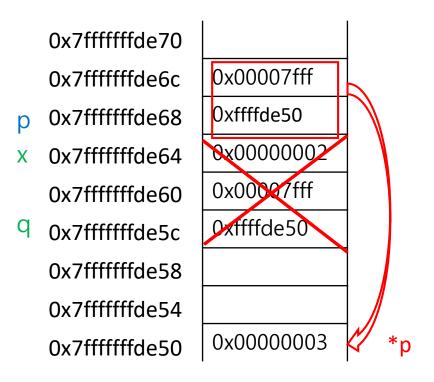
그러나 동적 할당받은 주소를 반환하는 것은 가능하다.

```
int *f(int x){
      int *q = new int;
      *q = x+1;
      return q;
 9
10
    int main(){
12
      int *p;
13
      p = f(2);
14
      cout << *p << endl;
16
```



그러나 동적 할당받은 주소를 반환하는 것은 가능하다.

```
int *f(int x){
         int *q = new int;
    6
         *q = x+1;
         return q;
    9
  10
       int main(){
  12
         int *p;
  13
      p = f(2);
\Rightarrow 14
      cout << *p << endl;
  15
  16 }
```



1차원 배열을 동적으로 할당

```
• 배열
double *pd = new double[10];
...
delete[] pd;
```

실습

• 3차원 배열을 만들어서 반환하는 함수 makeArray3D() 와 3차원 함수를 heap 에서 제거하는 함수 destroyArray3D() 를 완성하라.

```
#include <iostream>
    #include <cstdlib>
    using namespace std;
 4
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
    int main(int argc, char *argv[]){
      if (argc < 2){
9
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
        return -1:
11
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
19
      int *arrld = NULL;
20
      arrld = makeArray1D(size);
21
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
23
      for (int i=0; i<size[0]; i++) cout << arr1d[i] << " ";
      cout << endl;
24
      destroyArray1D(arr1d, size);
25
      return 0;
26
27 }
```

```
ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
0 1 2
```

```
#include <iostream>
    #include <cstdlib>
    using namespace std;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
18
19
      int *arr1d = NULL;
20
21
      arrld = makeArray1D(size);
      for (int i=0; i<size[0]; i++) arrld[i] = i;
23
      for (int i=0; i<size[0]; i++) cout << arr1d[i] << " ";
24
      cout << endl;
25
      destroyArray1D(arr1d, size);
                                       ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
26
      return 0;
27 }
```

dim	0x7fffffffde70	0x00000001	
	0x7fffffffde6c	0x00007fff	
size	0x7fffffffde68	0xffffde34	N
	0x7fffffffde64	0x00000000	
arr1d	0x7fffffffde60	0x00000000	
	0x7fffffffde5c		
	0x7fffffffde58		
	0x7fffffffde54		
	0x7fffffffde50		
	0x7fffffffde4c		
	0x7fffffffde48		
	0x7fffffffde44		
	0x7fffffffde40		
	0x7fffffffde3c		
	0x7fffffffde38		,
	0x7fffffffde34	0x00000003	4

```
int *makeArray1D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                  int n = sz[0];
    using namespace std;
                                                  int *arr = new int[n];
                                                  return arr;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
                                                void destroyArray1D(int *arr,int *sz){
                                           34
                                                  delete[] arr;
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
19
      int *arr1d = NULL;
20
      arrld = makeArray1D(size);
21
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
      for (int i=0; i<size[0]; i++) cout << arrld[i] << " ";
23
24
      cout << endl;
25
      destroyArray1D(arr1d, size);
      return 0;
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
26
27 }
```

dim	0x7ffffffde70	0x00000001	
	0x7fffffffde6c	0x00007fff	
size	0x7fffffffde68	0xffffde34	
	0x7fffffffde64	0x00000000	
arr1d	0x7fffffffde60	0x00000000	
	0x7fffffffde5c	0x00007fff	
SZ	0x7fffffffde58	0xffffde34	
n	0x7fffffffde54	0x00000003	
	0x7fffffffde50	0x00007fff	
arr	0x7fffffffde4c	0xffffde38	
	0x7fffffffde48		
	0x7fffffffde44		
	0x7fffffffde40		
	0x7fffffffde3c		
	0x7fffffffde38		K/
	0x7ffffffde34	0x00000003	// sz[0]

```
int *makeArray1D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                 int n = sz[0];
    using namespace std;
                                                  int *arr = new int[n];
                                                  return arr;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
                                               void destroyArray1D(int *arr,int *sz){
                                          34
                                                  delete[] arr;
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
17
19
      int *arr1d = NULL;
20
      arrld = makeArray1D(size);
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
      for (int i=0; i<size[0]; i++) cout << arrld[i] << " ";
23
24
      cout << endl;
25
      destroyArray1D(arr1d, size);
26
      return 0;
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
```

27 }

dim	0x7fffffffde70	0x0000001	
	0x7fffffffde6c	0x00007fff	l
size	0x7fffffffde68	0xffffde34	\mathcal{N}
	0x7fffffffde64	0x00007fff	\
arr1d	0x7fffffffde60	0xffffde38	
	0x7fffffffde5c	0x00007fff	
	0x7fffffffde58	0xffffde34	\\
	0x7fffffffde54	0x00000003	\
	0x7fffffffde50	0x00007ff	1
	0x7fffffffde4c	0xffffde38	
	0x7fffffffde48		
	0x7fffffffde44		
	0x7fffffffde40		1 //
	0x7fffffffde3c		
	0x7fffffffde38		K
	0x7ffffffde34	0x00000003	4

```
int *makeArray1D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                 int n = sz[0];
    using namespace std;
                                                  int *arr = new int[n];
                                                  return arr;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
                                                void destroyArray1D(int *arr,int *sz){
                                           34
                                                  delete[] arr;
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
17
19
      int *arr1d = NULL;
20
      arrld = makeArray1D(size);
21
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
      for (int i=0; i<size[0]; i++) cout << arrld[i] << " ";
23
24
      cout << endl;
      destroyArray1D(arr1d, size);
26
      return 0;
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
```

		_	_
dim	0x7fffffffde70	0x00000001	
	0x7fffffffde6c	0x00007fff	
size	0x7ffffffde68	0xffffde34	
	0x7fffffffde64	0x00007fff	
arr1d	0x7fffffffde60	0xffffde38	
	0x7fffffffde5c	0x00007fff	
	0x7fffffffde58	0xfffde34	
	0x7fffffffde54	0x00000003	
	0x7fffffffde50	0x00007ff	
	0x7fffffffde4c	0xffffde38	
	0x7fffffffde48		
	0x7fffffffde44		
	0x7fffffffde40	0x00000002	
	0x7fffffffde3c	0x00000001	
	0x7fffffffde38	0x00000000	
	0x7fffffffde34	0x00000003	4

1차원 배열 destroyArray1D()

```
int *makeArray1D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                 int n = sz[0];
    using namespace std;
                                                 int *arr = new int[n];
                                                  return arr;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
                                               void destroyArray1D(int *arr,int *sz){
                                                  delete[] arr;
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
      int *size = new int[dim];
15
16
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
17
19
      int *arr1d = NULL;
20
      arrld = makeArray1D(size);
21
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
      for (int i=0; i<size[0]; i++) cout << arr1d[i] << " ";
23
24
      cout << endl;
25
      destroyArray1D(arr1d, size);
      return 0;
26
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
27 }
```

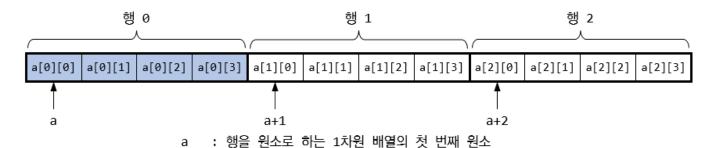
dim	0x7fffffffde70	0x00000001	
	0x7fffffffde6c	0x00007fff	
size	0x7fffffffde68	0xffffde34	
	0x7fffffffde64	0x00007fff	
arr1d	0x7fffffffde60	0xffffde38	
	0x7fffffffde5c	0x00007fff	K
arr	0x7fffffffde58	0xffffde38	
	0x7fffffffde54	0x00007fff] \
SZ	0x7fffffffde50	0xffffde34	
	0x7fffffffde4c		
	0x7fffffffde48		
	0x7fffffffde44		
	0x7fffffffde40	0x00000002	
	0x7fffffffde3c	0x00030001	
	0x7ffffffde38	0x00000000	
	0x7ffffffde34	0x00000003	4

1차원 배열 destroyArray1D()

```
int *makeArray1D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                  int n = sz[0];
    using namespace std;
                                                  int *arr = new int[n];
                                                  return arr;
    int *makeArray1D(int *sz);
    void destroyArray1D(int *arr,int *sz);
                                               void destroyArray1D(int *arr,int *sz){
                                           34
                                                  delete[] arr;
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
      int i, dim = argc-1;
14
15
      int *size = new int[dim];
16
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
17
19
      int *arr1d = NULL;
20
      arrld = makeArray1D(size);
21
22
      for (int i=0; i<size[0]; i++) arrld[i] = i;
      for (int i=0; i<size[0]; i++) cout << arrld[i] << " ";
23
24
      cout << endl;
25
      destroyArray1D(arr1d, size);
      return 0;
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc1d 3
                  delete[] size;
```

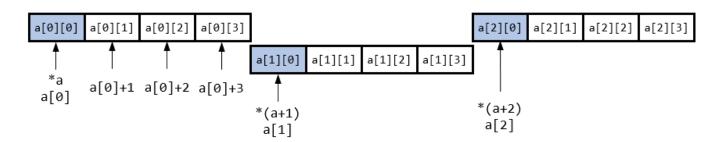
dim	0x7fffffffde70	0x00000001
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde34
	0x7fffffffde64	0x00007fff
arr1d	0x7fffffffde60	0xffffde40
	0x7fffffffde5c	0x00007fff
	0x7fffffffde58	0xffffde38
	0x7fffffffde54	0x0000Xfff
	0x7ffffffde50	0xffffde34
	0x7fffffffde4c	
	0x7fffffffde48	
	0x7fffffffde44	
	0x7fffffffde40	0×00000002
	0x7fffffffde3c	0x00030001
	0x7fffffffde38	0x00000000
	0x7fffffffde34	0x00000003
		1
		memory leak

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]



즉 첫 번째 행(원소가 4개인 1차원 배열)의 주소

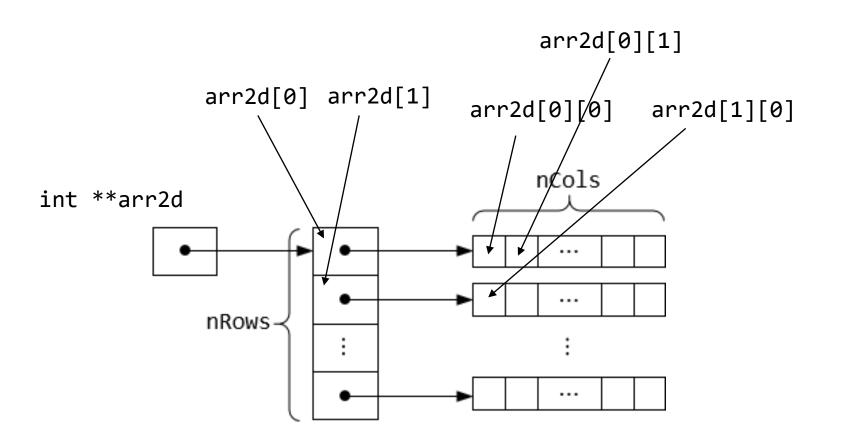
a+1 : 두 번째 행의 주소 a+2 : 세 번째 행의 주소



a[0], *a : 첫 번째 행(원소가 4개인 1차원 배열)의 첫 번째 원소 a[0][0] 의 주소

(따라서 첫 번째 행인 1차원 배열의 이름으로 사용할 수 있음)

a[1], *(a+1) : 두 번째 행의 첫 번째 원소 a[1][0]의 주소 (두 번째 행인 1차원 배열의 이름) a[2], *(a+2) : 세 번째 행의 첫 번째 원소 a[2][0]의 주소 (세 번째 행인 1차원 배열의 이름)



```
#include <iostream>
    #include <cstdlib>
    using namespace std;
    int **makeArray2D(int *sz);
    void destroyArray2D(int **arr,int *sz);
    int main(int argc, char *argv[]){
      if (argc < 2){
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
        return -1;
12
13
14
      int i, dim = argc-1;
15
      int *size = new int[dim];
16
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
      int **arr2d = NULL;
19
      arr2d = makeArray2D(size);
20
21
      for (int i=0; i<size[0]; i++)
22
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
      for (int i=0; i<size[0]; i++) {
23
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
         cout << endl;
25
      destroyArray2D(arr2d, size);
27
      return 0;
```

```
ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
0 1 2
3 4 5
```

```
#include <iostream>
    #include <cstdlib>
   using namespace std;
    int **makeArray2D(int *sz);
   void destroyArray2D(int **arr,int *sz);
    int main(int argc, char *argv[]){
     if (argc < 2){
9
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
11
        return -1;
12
13
14
      int i, dim = argc-1;
15
      int *size = new int[dim];
16
17
      for(i=1; i<arqc; i++) size[i-1] = atoi(arqv[i]);</pre>
      int **arr2d = NULL;
      arr2d = makeArray2D(size);
20
21
      for (int i=0; i<size[0]; i++)
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
      for (int i=0; i<size[0]; i++) {
23
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
25
         cout << endl;
26
                                             ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
      return 0;
```

```
dim 0x7ffffffde70| 0x00000002
      0x7ffffffde6c 0x00007fff
     0x7ffffffde68 0xffffde18
      0x7ffffffde64  0x00000000
arr2d 0x7ffffffde60 | 0x00000000
      0x7fffffffde5c
      0x7ffffffde58
      0x7fffffffde54
      0x7fffffffde50
      0x7fffffffde4c
      0x7fffffffde48
      0x7fffffffde44
      0x7fffffffde40
      0x7fffffffde3c
      0x7fffffffde38
      0x7ffffffde34
      0x7fffffffde30
      0x7fffffffde2c
      0x7fffffffde28
      0x7fffffffde24
      0x7fffffffde20
      0x7ffffffde1c | 0x00000003
      0x7ffffffde18 0x00000002
```

```
#include <iostream>
                                                       int **makeArray2D(int *sz){
    #include <cstdlib>
                                                         int n1 = sz[0], n2 = sz[1];
    using namespace std;
                                                  32 \longrightarrow int **arr = new int *[n1];
                                                         for (int i=0; i<n1; i++)
                                                  33
    int **makeArray2D(int *sz);
    void destroyArray2D(int **arr,int *sz);
                                                            arr[i] = new int[n2];
                                                  34
                                                                                                     SZ
                                                         return arr;
    int main(int argc, char *argv[]){
                                                                                                     n1
      if (argc < 2){
                                                       void destroyArray2D(int **arr,int *sz){
                                                  37
        cout << "usage : ./str 1d 2d 3d ... nd \n
10
                                                         int n1 = sz[0];
        return -1;
11
                                                  39
                                                         for (int i=0; i<n1; i++)
12
                                                                                                     arr
                                                            delete[] arr[i];
                                                  40
13
                                                  41
                                                         delete[] arr;
      int i, dim = argc-1;
14
      int *size = new int[dim];
15
16
17
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);
18
19
      int **arr2d = NULL;
20
      arr2d = makeArray2D(size);
21
      for (int i=0; i<size[0]; i++)
22
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
23
      for (int i=0; i<size[0]; i++) {
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
25
         cout << endl;
26
                                          ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
28
      return 0;
```

```
0x7fffffffde6c
                   0x00007fff
     0x7fffffffde68
                   0xffffde18
     0x7fffffffde64
                   0x00000000
arr2d0x7ffffffde60
                   0x00000000
     0x7fffffffde5c
                   0x00007fff
     0x7ffffffde58 0xffffde18
     0x7fffffffde54
                   0x00000002
     0x7ffffffde50 | 0x00000003
     0x7ffffffde4c 0x00007fff
     0x7ffffffde48 0xffffde20
     0x7fffffffde44
     0x7fffffffde40
     0x7fffffffde3c
     0x7fffffffde38
     0x7ffffffde34
     0x7fffffffde30
     0x7fffffffde2c
     0x7ffffffde28
     0x7fffffffde24
     0x7fffffffde20
     0x7fffffffde1c |
                   0x00000003
     0x7ffffffde18 0x00000002
```

dim 0x7ffffffde70

0x00000002

```
0x7ffffffde68
                                                                                                                           0xffffde18
                                                                                                         size
    #include <iostream>
                                                          int **makeArray2D(int *sz){
    #include <cstdlib>
                                                                                                              0x7fffffffde64
                                                                                                                           0x00000000
                                                            int n1 = sz[0], n2 = sz[1];
                                                    31
    using namespace std;
                                                            int **arr = new int *[n1];
                                                                                                         arr2d 0x7ffffffde60
                                                                                                                           0x00000000
                                                            for (int i=0; i<n1; i++)
                                                                                                              0x7fffffffde5c
                                                                                                                           0x00007fff
    int **makeArray2D(int *sz);
                                                    34
                                                               arr[i] = new int[n2];
    void destroyArray2D(int **arr,int *sz);
                                                                                                              0x7fffffffde58
                                                                                                                           0xffffde18
                                                                                                         SZ
                                                            return arr;
                                                                                                              0x7fffffffde54
                                                                                                                           0x00000002
    int main(int argc, char *argv[]){
      if (argc < 2){
                                                                                                              0x7fffffffde50
                                                    37
                                                         void destroyArray2D(int **arr,int *sz){
                                                                                                                           0x0000003
10
        cout << "usage : ./str 1d 2d 3d ... nd \n'
                                                            int n1 = sz[0];
                                                                                                              0x7fffffffde4c 0x00007fff
11
        return -1;
                                                    39
                                                            for (int i=0; i<n1; i++)
12
                                                                                                              0x7ffffffde48 0xffffde20
                                                                                                         arr
                                                               delete[] arr[i];
                                                    40
13
                                                                                                              0x7fffffffde44
                                                            delete[] arr;
                                                    41
14
      int i, dim = argc-1;
                                                                                                              0x7fffffffde40
      int *size = new int[dim];
15
16
                                                                                                              0x7fffffffde3c
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
                                                                                                              0x7fffffffde38
18
19
      int **arr2d = NULL;
                                                                                                              0x7fffffffde34
      arr2d = makeArray2D(size);
20
                                                                                                              0x7fffffffde30
      for (int i=0; i<size[0]; i++)
21
                                                                                                              0x7ffffffde2c
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
      for (int i=0; i<size[0]; i++) {
23
                                                                                                         arr[1] 0x7ffffffde28
24
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
                                                                                                              0x7fffffffde24
                                                                                                                           0x00007fff
25
         cout << endl;
26
                                                                                                         arr[0] 0x7ffffffde20
                                                                                                                           0xffffde30
                                            ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
                                                                                                              0x7fffffffde1c
                                                                                                                           0x00000003
      return 0:
                                                                                                              0x7ffffffde18 0x00000002
```

0x7fffffffde70

0x7fffffffde6c

0x00000002

0x00007fff

```
0x7ffffffde68
                                                                                                                           0xffffde18
                                                                                                         lsize
                                                         int **makeArray2D(int *sz){
    #include <iostream>
    #include <cstdlib>
                                                            int n1 = sz[0], n2 = sz[1];
                                                                                                              0x7fffffffde64
                                                                                                                           0x00000000
                                                    31
    using namespace std;
                                                    32
                                                            int **arr = new int *[n1];
                                                                                                         arr2d 0x7fffffffde60
                                                                                                                           0x00000000
                                                            for (int i=0; i<n1; i++)
                                                    33
    int **makeArray2D(int *sz);
                                                                                                              0x7fffffffde5c
                                                                                                                           0x00007fff
                                                         arr[i] = new int[n2];
                                                    34
    void destroyArray2D(int **arr,int *sz);
                                                                                                              0x7ffffffde58 0xffffde18
                                                    35
                                                            return arr;
                                                                                                              0x7fffffffde54
                                                                                                                           0x00000002
    int main(int argc, char *argv[]){
      if (argc < 2){
                                                    37
                                                          void destroyArray2D(int **arr,int *sz){
                                                                                                              0x7fffffffde50
                                                                                                                           0x0000003
10
        cout << "usage : ./str 1d 2d 3d ... nd \n'</pre>
                                                            int n1 = sz[0];
                                                                                                              0x7fffffffde4c
                                                                                                                           0x00007fff
11
        return -1;
                                                            for (int i=0; i<n1; i++)
                                                    39
12
                                                                                                              0x7fffffffde48
                                                                                                                           0xffffde20
                                                                                                         larr
                                                    40
                                                               delete[] arr[i];
13
                                                                                                              0x7fffffffde44
                                                            delete[] arr;
                                                    41
      int i, dim = argc-1;
14
                                                                                                              0x7fffffffde40
15
      int *size = new int[dim];
16
                                                                                                              0x7fffffffde3c
17
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
                                                                                                              0x7fffffffde38
18
19
      int **arr2d = NULL;
                                                                                                              0x7fffffffde34
      arr2d = makeArray2D(size);
20
                                                                                                              0x7fffffffde30
      for (int i=0; i<size[0]; i++)
21
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
22
                                                                                                              0x7ffffffde2c
                                                                                                                           0x00007fff
23
      for (int i=0; i<size[0]; i++) {
                                                                                                                           0xffffde3c
                                                                                                         arr[1] 0x7fffffffde28
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
                                                                                                              0x7fffffffde24
                                                                                                                           0x00007fff
25
         cout << endl;</pre>
26
                                                                                                         arr[0] 0x7ffffffde20
                                                                                                                           0xffffde30
                                            ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
                                                                                                              0x7fffffffde1c
                                                                                                                           0x00000003
28
      return 0;
                                                                                                              0x7ffffffde18 0x00000002
```

0x7fffffffde70

0x7fffffffde6c

0x00000002

0x00007fff

#include <iostream>

```
#include <cstdlib>
                                                        int **makeArray2D(int *sz){
                                                                                                               0x7fffffffde64
                                                                                                                            0x00007fff
    using namespace std;
                                                          int n1 = sz[0], n2 = sz[1];
                                                   31
                                                                                                               0x7fffffffde60
                                                                                                                            0xfffde20
                                                                                                        arr2d
                                                          int **arr = new int *[n1];
    int **makeArray2D(int *sz);
                                                                                                               0x7fffffffde5c 0x00007fff
                                                          for (int i=0; i<n1; i++)
    void destroyArray2D(int **arr,int *sz);
                                                   34
                                                             arr[i] = new int[n2];
                                                                                                               0x7ffffffde58
                                                                                                                            0xffffde18
                                                          return arr;
    int main(int argc, char *argv[]){
                                                                                                               0x7fffffffde54
                                                                                                                            0x00000002
      if (argc < 2){
                                                        void destroyArray2D(int **arr,int *sz){
                                                                                                                            0x00000003
                                                                                                               0x7fffffffde50
10
        cout << "usage : ./str 1d 2d 3d ... nd \
                                                          int n1 = sz[0];
                                                                                                               0x7ffffffde4c | 0x00007fff
        return -1;
                                                   39
                                                          for (int i=0; i<n1; i++)
12
                                                                                                               0x7ffffffde48 0xffffde20
                                                             delete[] arr[i];
13
                                                   41
                                                          delete[] arr;
                                                                                                               0x7ffffffde44 0x00000005
                                                                                                   arr2d[1][2]
14
      int i, dim = argc-1;
      int *size = new int[dim];
15
                                                                                                               0x7ffffffde40 0x00000004
                                                                                                   arr2d[1][1]
16
                                                                                                               0x7ffffffde3c 0x00000003
                                                                                                   arr2d[1][0]
17
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
                                                                                                               0x7ffffffde38 0x00000002
                                                                                                   arr2d[0][2]
18
      int **arr2d = NULL;
19
                                                                                                               0x7ffffffde34 0x00000001
                                                                                                   arr2d[0][1]
20
      arr2d = makeArray2D(size);
                                                                                                               0x7ffffffde30 0x00000000
                                                                                                   arr2d[0][0]
      for (int i=0; i<size[0]; i++)
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
                                                                                                               0x7fffffffde2c
                                                                                                                            0x00007fff
23
      for (int i=0; i<size[0]; i++) {
                                                                                                               0x7fffffffde28
                                                                                                                            0xffffde3c
                                                                                                      arr2d[1]
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
                                                                                                               0x7fffffffde24
                                                                                                                            0x00007fff
         cout << endl;
26
                                                                                                               0x7fffffffde20
                                                                                                                            0xffffde30
                                                                                                      arr2d[0]
                                       ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
      destroyArray2D(arr2d, size);
27
                                                                                                               0x7fffffffde1c
                                                                                                                            0x00000003
28
      return 0;
29 }
                                                                                                               0x7ffffffde18 0x00000002
```

0x7fffffffde70

0x7fffffffde6c

0x7ffffffde68

0x00000002

0x00007fff

0xffffde18

2차원 배열 destroyArray2D()

```
#include <iostream>
                                                      int **makeArray2D(int *sz){
    #include <cstdlib>
                                                 31
                                                        int n1 = sz[0], n2 = sz[1];
    using namespace std;
                                                        int **arr = new int *[n1];
                                                        for (int i=0; i<n1; i++)
    int **makeArray2D(int *sz);
                                                 34
                                                           arr[i] = new int[n2];
    void destroyArray2D(int **arr,int *sz);
                                                        return arr;
    int main(int argc, char *argv[]){
                                                      void destroyArray2D(int **arr,int *sz){
      if (argc < 2){
10
        cout << "usage : ./str 1d 2d 3d ... nd \ 38
                                                        int n1 = sz[0];
11
        return -1;
                                                 39
                                                        for (int i=0; i<n1; i++)
12
                                                 40
                                                           delete[] arr[i];
13
                                                 41
                                                        delete[] arr;
      int i, dim = argc-1;
14
      int *size = new int[dim];
15
16
17
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
18
19
      int **arr2d = NULL;
      arr2d = makeArray2D(size);
20
      for (int i=0; i<size[0]; i++)
21
22
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
23
      for (int i=0; i<size[0]; i++) {
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
         cout << endl;</pre>
25
26
                                     ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
28
      return 0;
```

```
0x7fffffffde6c
                     0x00007fff
      0x7ffffffde68
                     0xffffde18
       0x7fffffffde64
                     0x00007fff
      0x7fffffffde60
                     0xfffde20
arr2d
       0x7fffffffde5c
                    0x00007fff
      0x7ffffffde58 0xffffde20
                    0x00007fff
       0x7fffffffde54
      0x7fffffffde50
                     0xffffde18
       0x7fffffffde4c
                     0x00000002
       0x7fffffffde48
       0x7ffffffde44 0x00000000
       0x7ffffffde40 0x00 0004
       0x7ffffffde3c 0x0000000
       0x7ffffffde38 0x00000000
       0x7ffffffde34 0x00 0001
       0x7ffffffde30 0x00000000
       0x7fffffffde2c
                     0x00007fff
       0x7fffffffde28
                    0xffffde3c
arr[1]
       0x7fffffffde24
                     0x00007fff
       0x7fffffffde20
                     0xffffde30
arr[0]
       0x7fffffffde1c
                    0x00000003
```

0x7ffffffde18 0x00000002

0x7fffffffde70

0x00000002

2차원 배열 destroyArray2D()

```
#include <iostream>
                                                      int **makeArray2D(int *sz){
                                                                                                          0x7fffffffde68
                                                                                                                       0xffffde18
                                                                                                     size
    #include <cstdlib>
                                                        int n1 = sz[0], n2 = sz[1];
                                                 31
                                                                                                           0x7fffffffde64
                                                                                                                       0x00007fff
    using namespace std;
                                                        int **arr = new int *[n1];
                                                 32
                                                                                                          0x7fffffffde60
                                                                                                                       0xfffde20
                                                                                                    arr2d
                                                        for (int i=0; i<n1; i++)
                                                 33
    int **makeArray2D(int *sz);
                                                                                                           0x7ffffffde5c
                                                                                                                       0x00007fff
                                                 34
                                                            arr[i] = new int[n2];
    void destroyArray2D(int **arr,int *sz);
                                                        return arr;
                                                                                                          0x7ffffffde58 0xffffde20
    int main(int argc, char *argv[]){
                                                                                                           0x7fffffffde54
                                                                                                                       0x00007fff
      if (argc < 2){
                                                      void destroyArray2D(int **arr,int *sz){
9
                                                                                                          0x7fffffffde50
                                                                                                                       0xffffde18
10
        cout << "usage : ./str 1d 2d 3d ... nd
                                                        int n1 = sz[0];
                                                                                                          0x7fffffffde4c
11
        return -1;
                                                                                                                       0x00000002
                                                        for (int i=0; i<n1; i++)
12
                                                 40
                                                            delete[] arr[i];
                                                                                                           0x7fffffffde48
13
                                                 41 delete[] arr;
                                                                                                           0x7ffffffde44 0x00000000
14
      int i, dim = argc-1;
                                                 42
15
      int *size = new int[dim];
                                                                                                           0x7ffffffde40 0x00 0004
16
                                                                                                           0x7ffffffde3c 0000000
17
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
                                                                                                           0x7ffffffde38
18
19
      int **arr2d = NULL;
                                                                                                           0x7ffffffde34 0x00 0001
      arr2d = makeArray2D(size);
20
                                                                                                           0x7ffffffde30|| 0x000000
      for (int i=0; i<size[0]; i++)
21
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
22
                                                                                                           0x7ffffffde2c 0x00007fff
23
      for (int i=0; i<size[0]; i++) {
                                                                                                          0x7ffffffde28 0xffffde3c
                                                                                                    arr[1]
24
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
                                                                                                           0x7ffffffde24 0x00007fff
         cout << endl;
25
26
                                                                                                          0x7ffffffde20 0xffffde30
                                                                                                    arr[0]
                                      ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
27
      destroyArray2D(arr2d, size);
                                                                                                           0x7ffffffde1c | 0x00000003
28
      return 0;
                                                                                                           0x7ffffffde18 0x00000002
```

0x7fffffffde70

0x7fffffffde6c

0x00000002

0x00007fff

2차원 배열 destroyArray2D()

```
int **makeArray2D(int *sz){
    #include <iostream>
                                                                                                      size
    #include <cstdlib>
                                                        int n1 = sz[0], n2 = sz[1];
                                                 31
                                                                                                           0x7fffffffde64
    using namespace std;
                                                 32
                                                        int **arr = new int *[n1];
                                                                                                           0x7fffffffde60
                                                                                                    arr2d
                                                 33
                                                        for (int i=0; i<n1; i++)
    int **makeArray2D(int *sz);
                                                 34
                                                            arr[i] = new int[n2];
    void destroyArray2D(int **arr,int *sz);
                                                 35
                                                         return arr;
    int main(int argc, char *argv[]){
                                                      void destroyArray2D(int **arr,int *sz){
      if (argc < 2){
                                                        int n1 = sz[0];
10
        cout << "usage : ./str 1d 2d 3d ... nd \ 38
11
        return -1;
                                                        for (int i=0; i<n1; i++)
                                                 39
12
                                                            delete[] arr[i];
                                                                                                           0x7fffffffde48
                                                 40
13
                                                 41
                                                        delete[] arr;
14
      int i, dim = argc-1;
                                                 42 }
15
      int *size = new int[dim];
16
17
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]);
18
19
      int **arr2d = NULL;
20
      arr2d = makeArray2D(size);
21
      for (int i=0; i<size[0]; i++)
         for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
22
23
      for (int i=0; i<size[0]; i++) {
                                                                                                    arr[1]
         for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
24
         cout << endl;</pre>
25
26
                                                                                                    arr[0]
27
      destroyArray2D(arr2d, size);
return 0;
                                                                                       memory
                 delete[] size;
                                                                                       loak
```

0x7fffffffde6c 0x00007fff 0x7ffffffde68 0xffffde18 0x00007fff 0xfffde20 0x7ffffffde5c \Qx00007fff 0x7ffffffde58 0xfffde20 $0x7fffffffde54 \mid 0x00007fff$ 0x7ffffffde50 | 0xffffde 8 0x7ffffffde4c / 0x00000000x7ffffffde44 0x00000000 0x7ffffffde40 0x00 0004 0x7ffffffde3c 0x000000 0x7ffffffde38 0x0000000 0x7ffffffde34 0x00 0001 0x7ffffffde30 0x000000 0x7ffffffde2c 0x00007fff 0x7ffffffde28 0xffffde3c 0x7fffffffde24 0x00007fff0x7ffffffde20 0xffffde30 0x7ffffffde1c | 0x00000003 0x7ffffffde18 0x00000002

0x7fffffffde70

0x00000002

2차원 배열: wrong example

```
int *x makeArray2D(int *sz);
    int main(int argc, char *argv[]){
      if (argc < 2){
         cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
10
         return -1;
11
12
      int i, dim = argc-1;
13
      int *size = new int[dim];
14
15
16
      for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);</pre>
17
      int *arr2d = NULL:
18
19
      arr2d = x makeArray2D(size);
      for (int i=0; i<size[0]; i++)
20
              for (int j=0; j<size[1]; j++) arr2d[i][j] = i*size[1]+j;
      for (int i=0; i<size[0]; i++) {
22
23
              for (int j=0; j<size[1]; j++) cout << arr2d[i][j] << ' ';
             cout << endl;
24
26
      return 0;
```

```
int *x_makeArray2D(int *sz){
int n1 = sz[0], n2 = sz[1];
int *arr = new int[n1*n2];
return arr;
}
```

```
int ***arr3d = NULL;
    #include <iostream>
                                                       20
    #include <cstdlib>
                                                              arr3d = makeArray3D(size);
    using namespace std;
                                                              for (int i=0; i<size[0]; i++)
4
    int ***makeArray3D(int *sz);
                                                                 for (int j=0; j<size[1]; j++)
                                                                     for (int k=0; k<size[2]; k++)
    void destroyArray3D(int ***arr,int *sz);
                                                        24
                                                                         arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
    int main(int argc, char *argv[]){
                                                        26
                                                              for (int i=0; i<size[0]; i++) {
      if (argc < 2){
                                                                 cout << "i : " << i << endl:
9
        cout << "usage : ./str 1d 2d 3d ... nd \n";</pre>
                                                                 for (int j=0; j<size[1]; j++){
                                                       28
        return -1;
11
                                                                     for (int k=0; k<size[2]; k++)
                                                        29
                                                                        cout << arr3d[i][j][k] << ' ';
13
                                                                     cout << endl:
14
      int i, dim = argc-1;
15
      int *size = new int[dim];
                                                                 cout << endl;
16
                                                       34
      for(i=1; i < argc; i++) size[i-1] = atoi(argv[i]); 35
17
                                                              destroyArray3D(arr3d, size);
                                                              return 0;
```

```
int ***arr3d = NULL:
20
      arr3d = makeArray3D(size);
21
      for (int i=0; i<size[0]; i++)
                                                              1
         for (int j=0; j<size[1]; j++)
                                                              3
             for (int k=0; k<size[2]; k++)
24
                arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
      for (int i=0; i<size[0]; i++) {
26
         cout << "i : " << i << endl;
27
28
         for (int j=0; j<size[1]; j++){
                                                           8 9
             for (int k=0; k<size[2]; k++)
29
                cout << arr3d[i][j][k] << ' ';
31
             cout << endl:
32
33
         cout << endl;
34
      destroyArray3D(arr3d, size);
      return 0;
37
```

```
ejim@ejim-VirtualBox:~/C2020$ ./alloc3d 3 4 2
i : 0
10 11
12 13
14 15
16 17
18 19
20 21
22 23
```

실습

- 3차원 배열을 만들어서 반환하는 함수 makeArray3D() 와 3차원 함수를 heap 에서 제거하는 함수 destroyArray3D() 를 완성하라.
- 제출 방법 : ecampus 의 숙제 제출 text 내용을 e-campus editor 에 입력 (copy & paste 사용) screen capture 제출은 안 됨

```
19
      int ***arr3d = NULL:
20
      arr3d = makeArray3D(size);
21
22
      for (int i=0; i<size[0]; i++)
23
         for (int j=0; j<size[1]; j++)
24
             for (int k=0; k<size[2]; k++)
                 arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
25
      for (int i=0; i<size[0]; i++) {
26
         cout << "i : " << i << endl:
27
         for (int j=0; j<size[1]; j++){
28
29
             for (int k=0; k<size[2]; k++)
                 cout << arr3d[i][j][k] << ' ';
             cout << endl;</pre>
31
         cout << endl;
34
35
      destroyArray3D(arr3d, size);
      return 0;
37
```

```
ejim@ejim-VirtualBox:~/C2020$ ./alloc3d 3 4 2
0 1
2 3
4 5
6 7
i : 1
8 9
10 11
12 13
14 15
i : 2
16 17
18 19
20 21
22 23
```