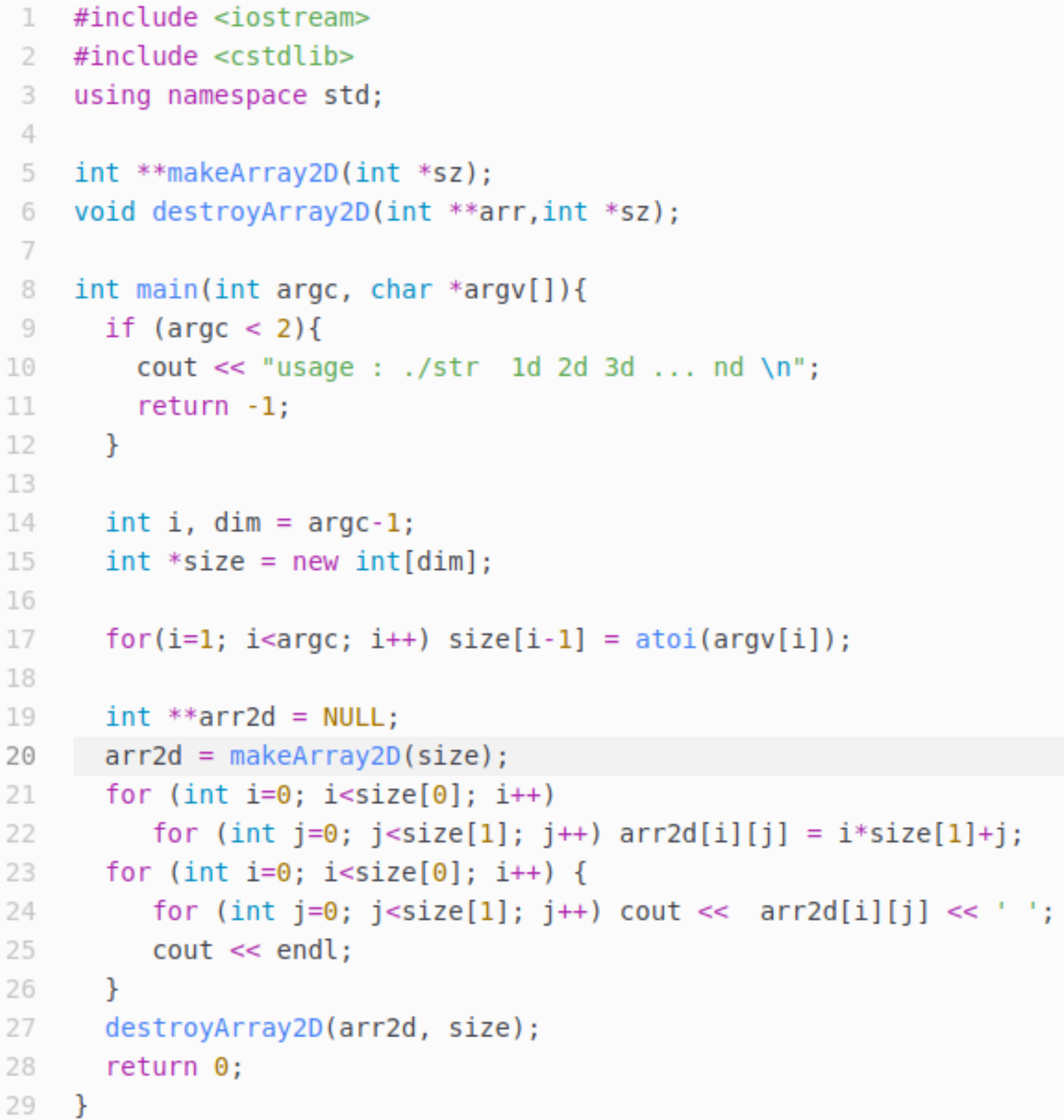
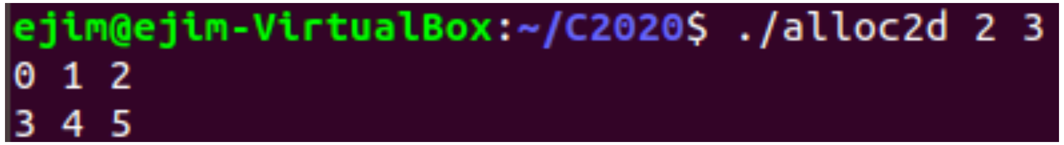
메모리의 동적 할당
dynamic allocation

국민대학교 임은진

2차원 배열 만들어 반환하기 makeArray2D()



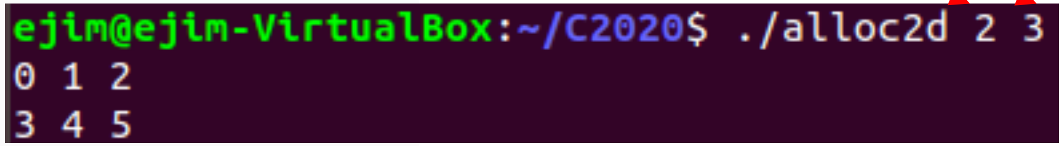
```cpp
#include <iostream>
#include <cstdlib>
using namespace std;

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";
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```



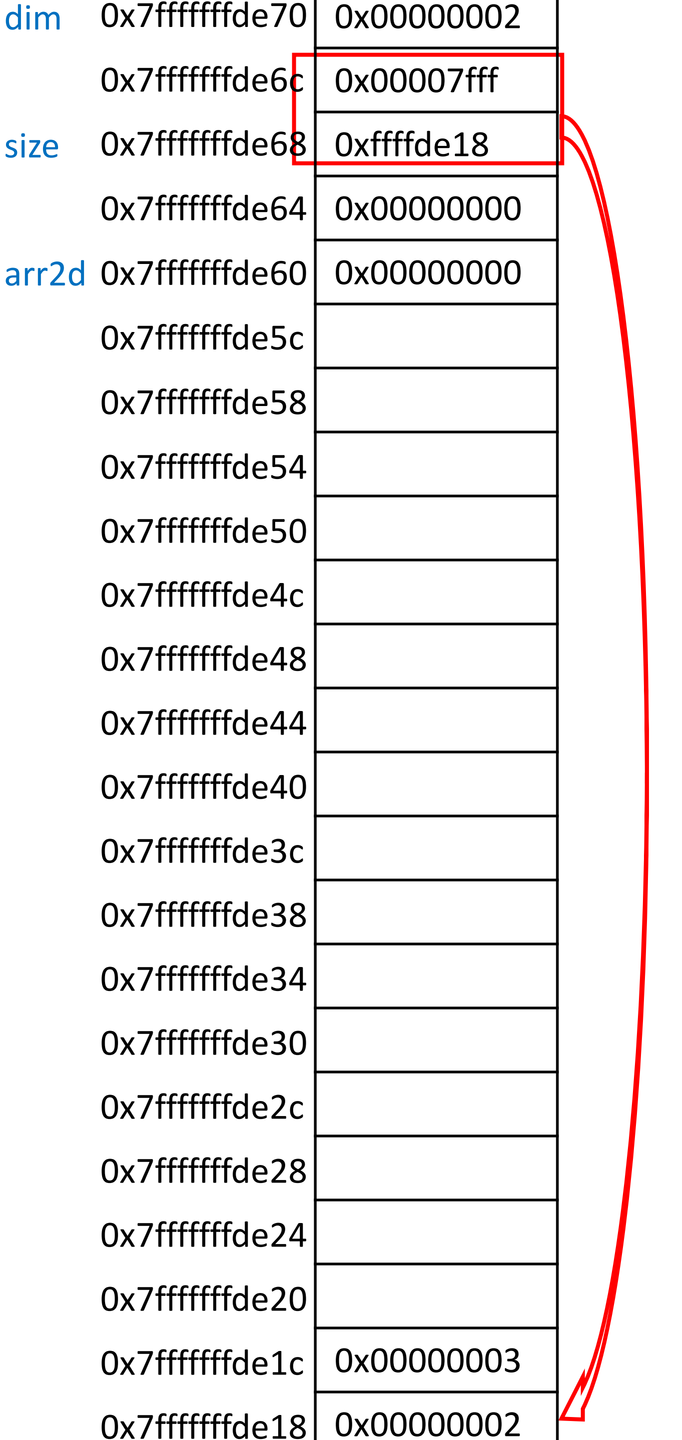
```
ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
0 1 2
3 4 5
```

2차원 배열 만들어 반환하기 makeArray2D()

```
#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";
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

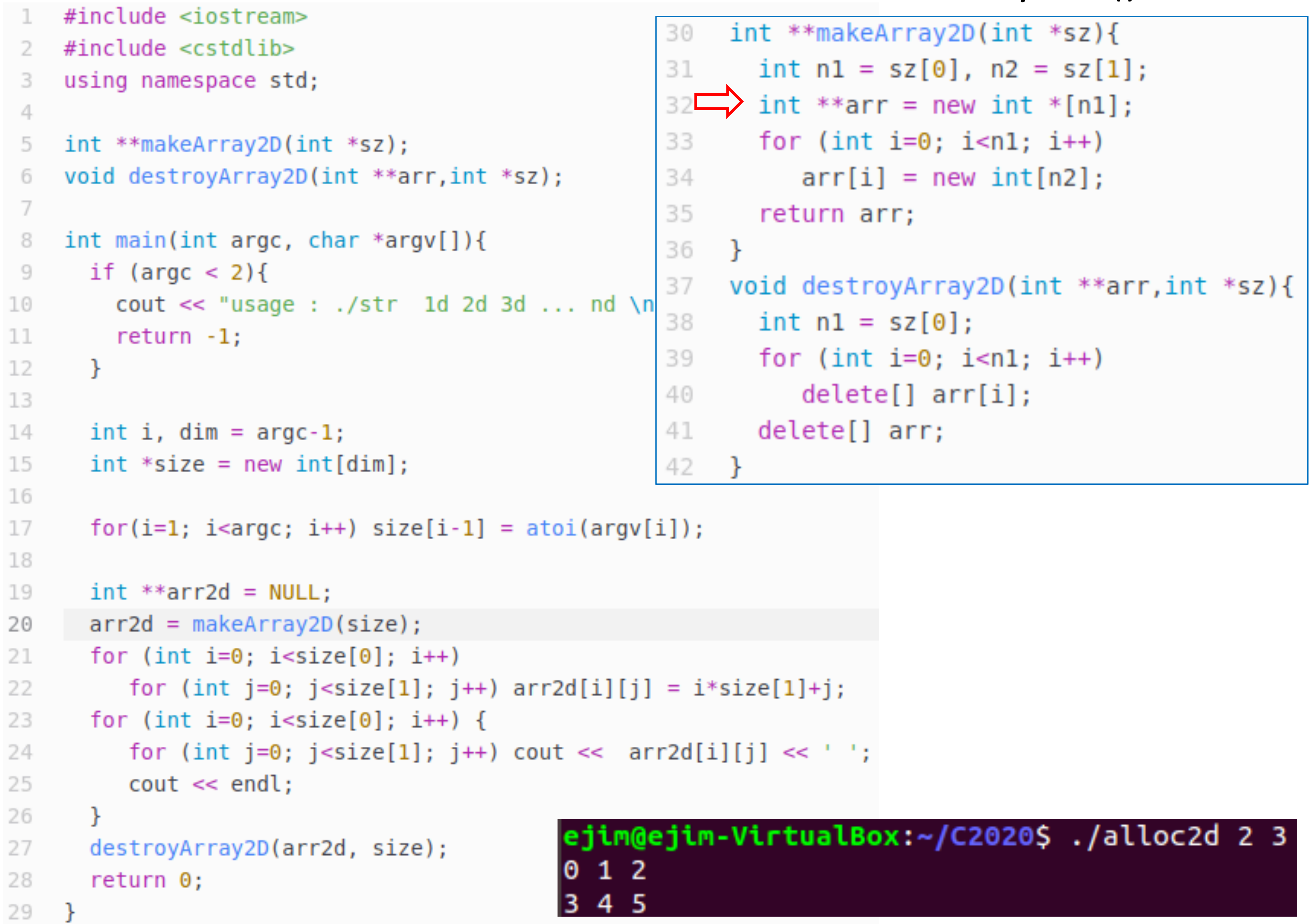


```
ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
0 1 2
3 4 5
```



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00000000
arr2d	0x7fffffffde60	0x00000000
	0x7fffffffde5c	
	0x7fffffffde58	
	0x7fffffffde54	
	0x7fffffffde50	
	0x7fffffffde4c	
	0x7fffffffde48	
	0x7fffffffde44	
	0x7fffffffde40	
	0x7fffffffde3c	
	0x7fffffffde38	
	0x7fffffffde34	
	0x7fffffffde30	
	0x7fffffffde2c	
	0x7fffffffde28	
	0x7fffffffde24	
	0x7fffffffde20	
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

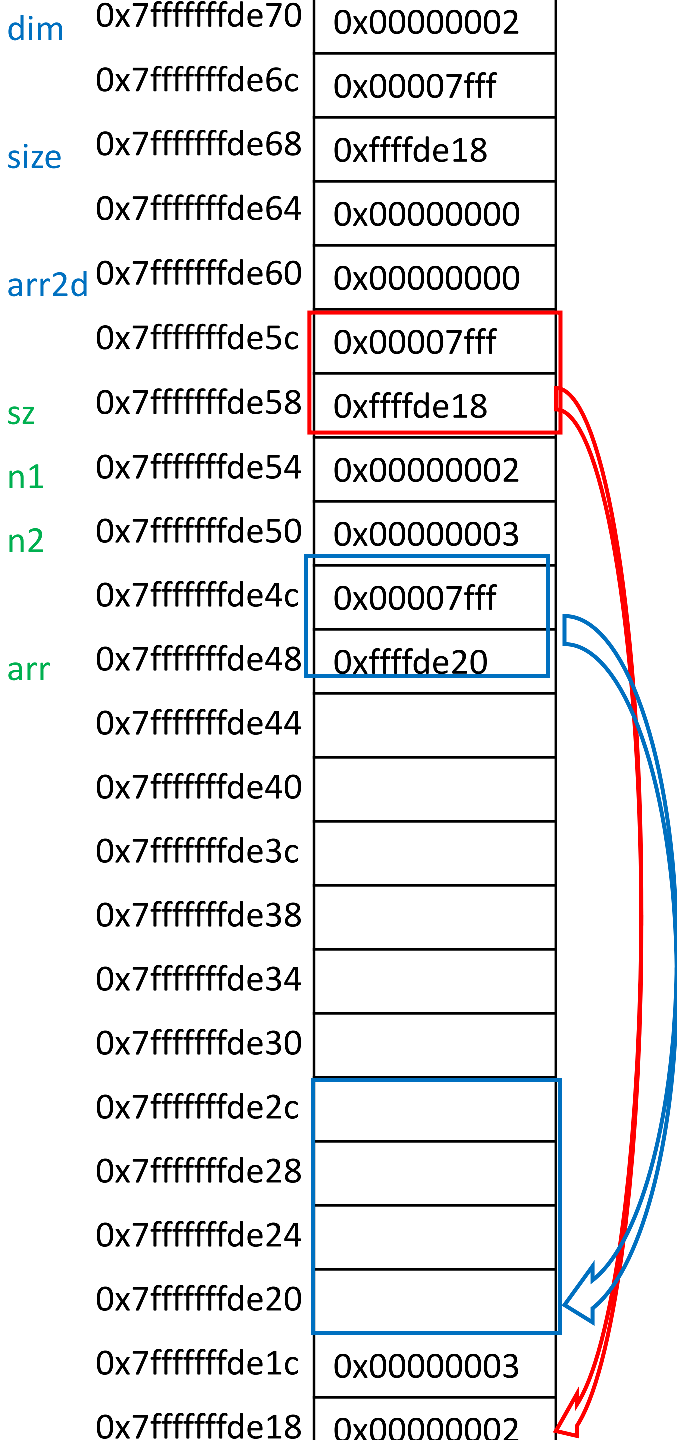
2차원 배열 만들어 반환하기 makeArray2D()



```cpp
#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
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

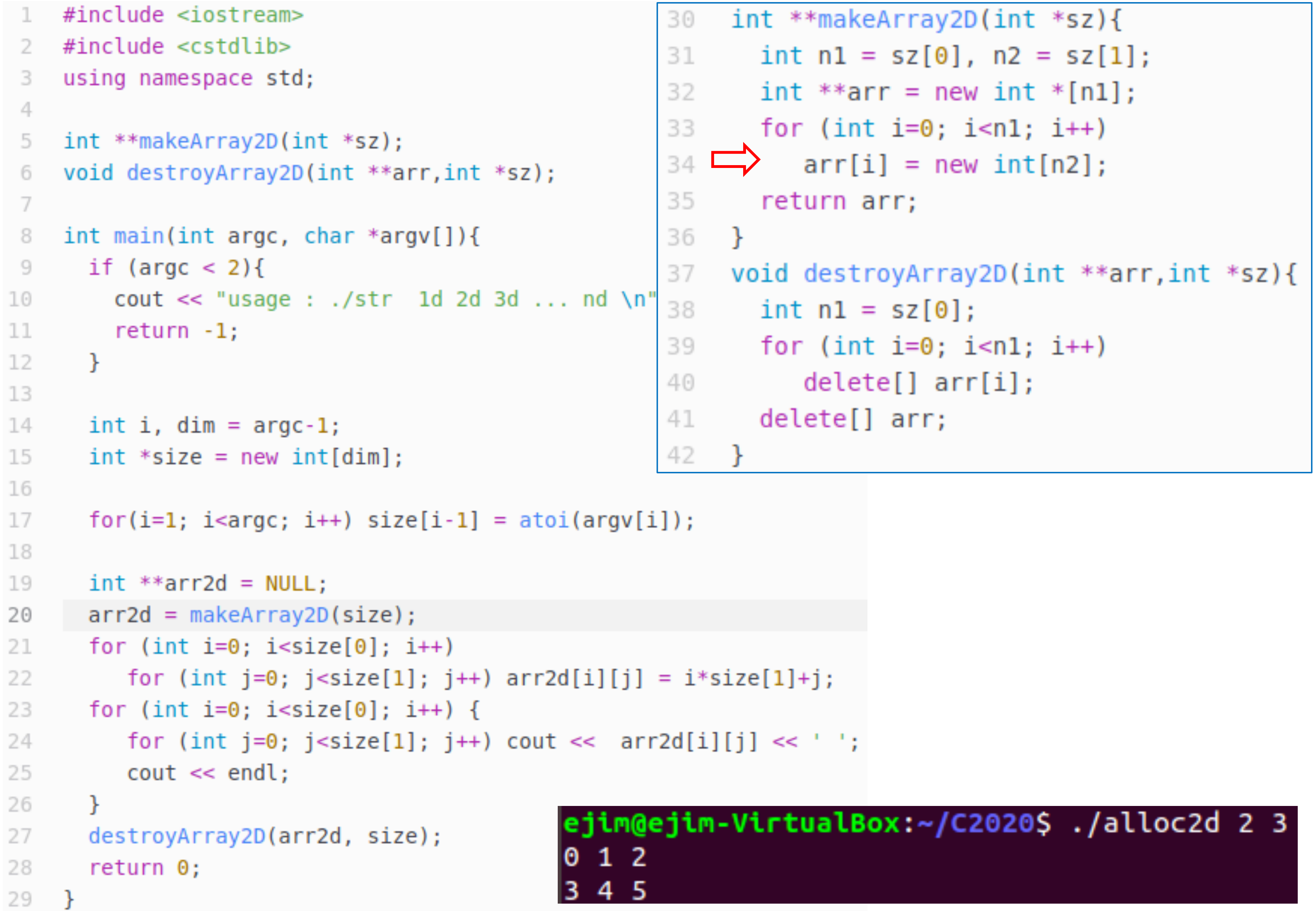
```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

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



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00000000
arr2d	0x7fffffffde60	0x00000000
	0x7fffffffde5c	0x00007fff
sz	0x7fffffffde58	0xffffde18
n1	0x7fffffffde54	0x00000002
n2	0x7fffffffde50	0x00000003
	0x7fffffffde4c	0x00007fff
arr	0x7fffffffde48	0xffffde20
	0x7fffffffde44	
	0x7fffffffde40	
	0x7fffffffde3c	
	0x7fffffffde38	
	0x7fffffffde34	
	0x7fffffffde30	
	0x7fffffffde2c	
	0x7fffffffde28	
	0x7fffffffde24	
	0x7fffffffde20	
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

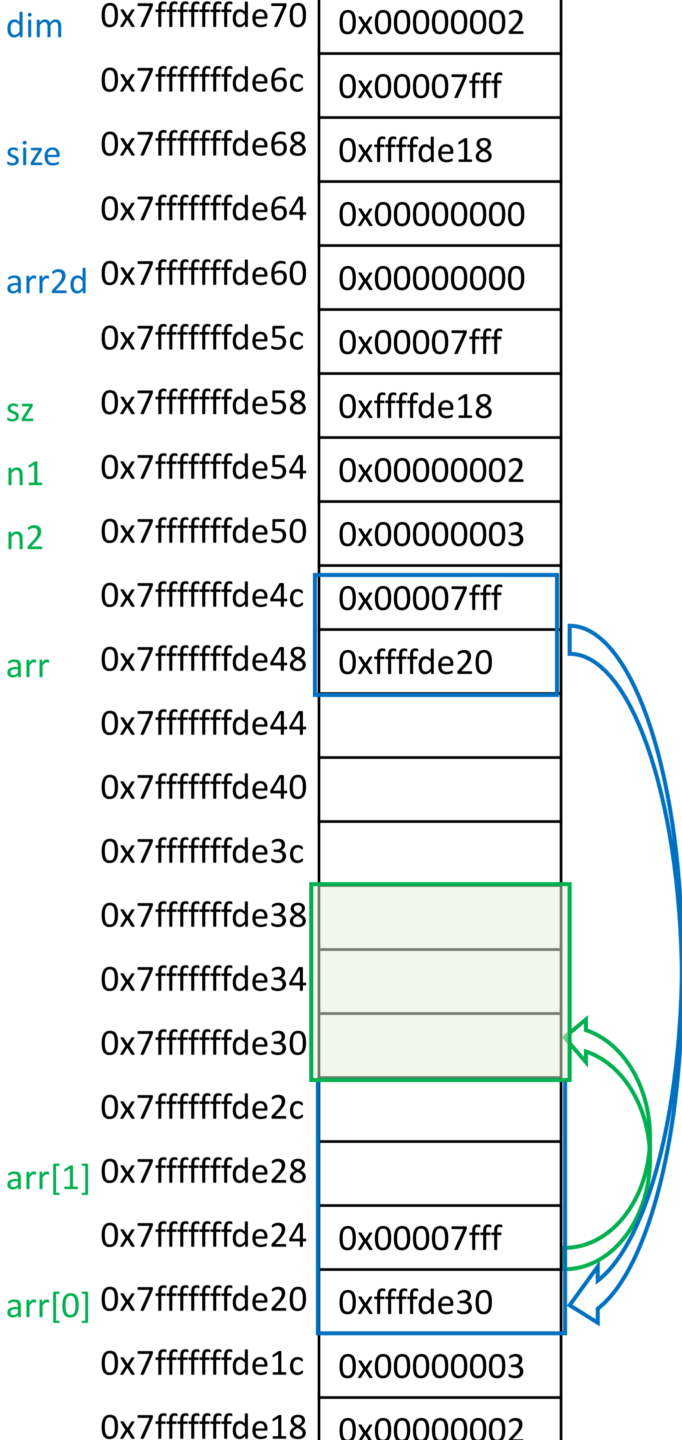
2차원 배열 만들어 반환하기 makeArray2D()



```cpp
#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"
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

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

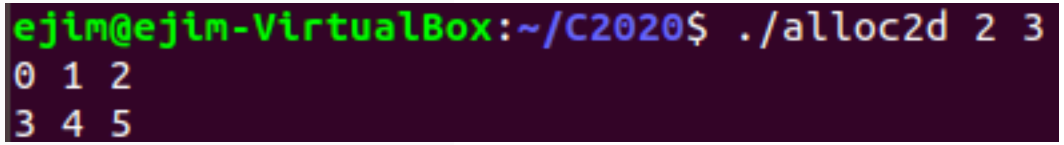


dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00000000
arr2d	0x7fffffffde60	0x00000000
	0x7fffffffde5c	0x00007fff
sz	0x7fffffffde58	0xffffde18
n1	0x7fffffffde54	0x00000002
n2	0x7fffffffde50	0x00000003
	0x7fffffffde4c	0x00007fff
arr	0x7fffffffde48	0xffffde20
	0x7fffffffde44	
	0x7fffffffde40	
	0x7fffffffde3c	
	0x7fffffffde38	
	0x7fffffffde34	
	0x7fffffffde30	
	0x7fffffffde2c	
arr[1]	0x7fffffffde28	
	0x7fffffffde24	0x00007fff
arr[0]	0x7fffffffde20	0xffffde30
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

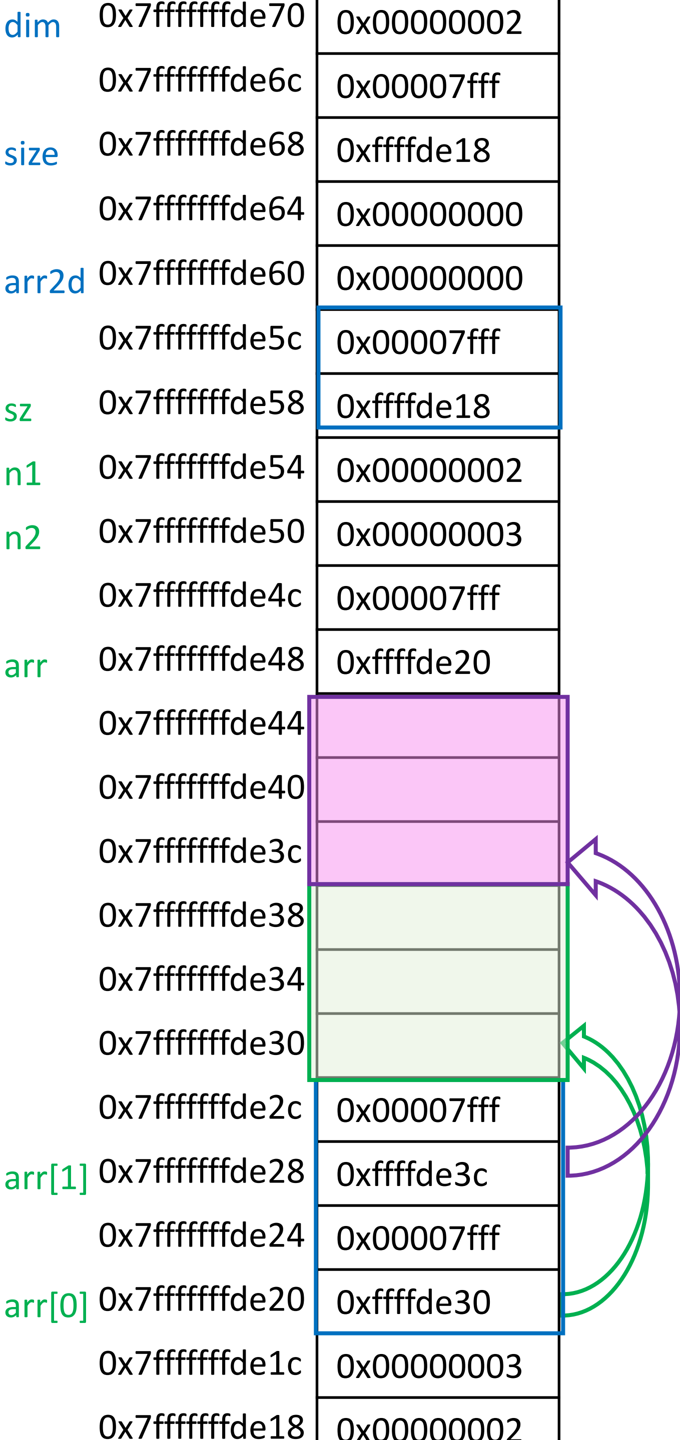
2차원 배열 만들어 반환하기 makeArray2D()

```cpp
#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
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

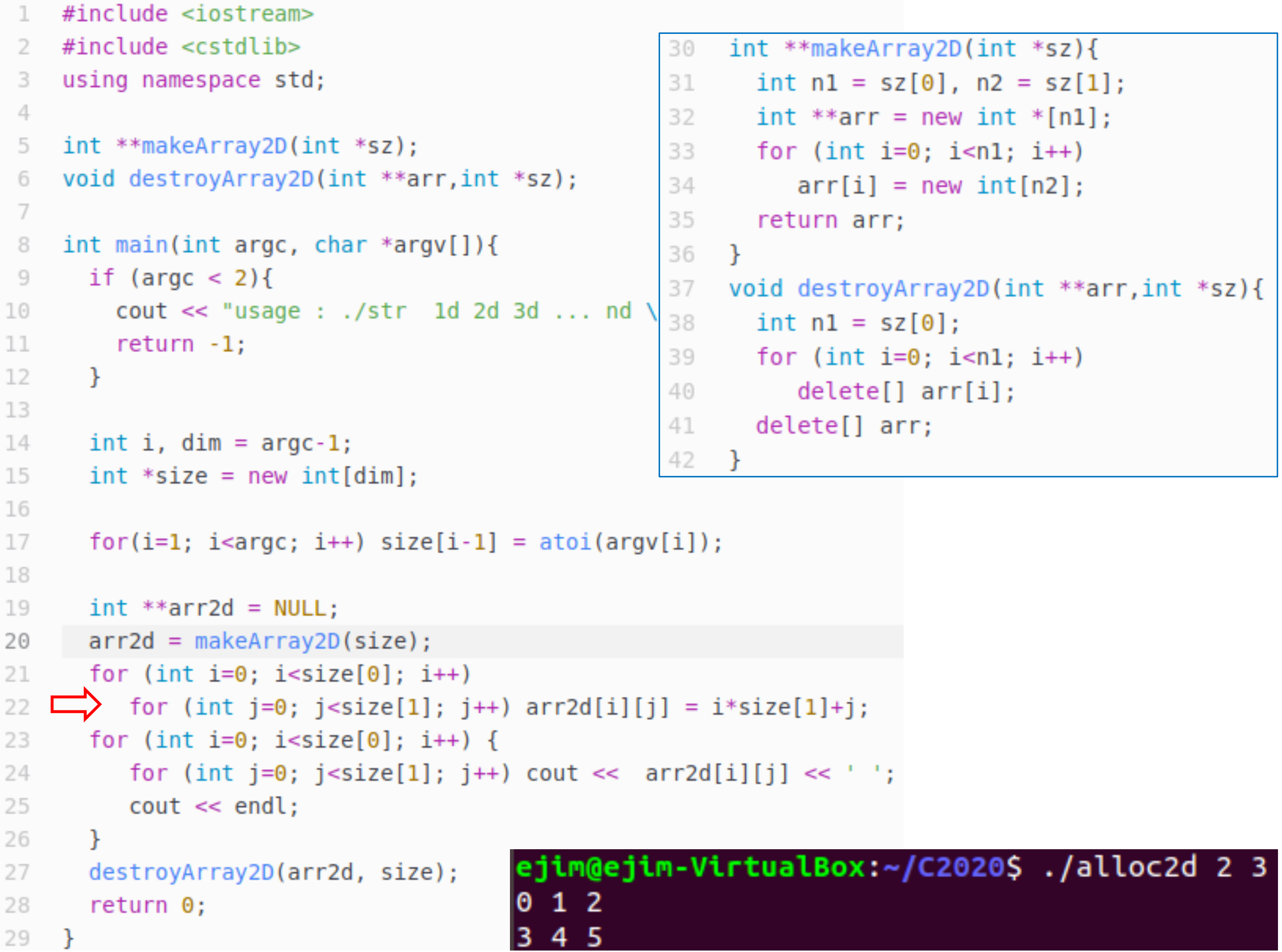


```
ejim@ejim-VirtualBox:~/C2020$ ./alloc2d 2 3
0 1 2
3 4 5
```



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00000000
arr2d	0x7fffffffde60	0x00000000
	0x7fffffffde5c	0x00007fff
sz	0x7fffffffde58	0xffffde18
n1	0x7fffffffde54	0x00000002
n2	0x7fffffffde50	0x00000003
	0x7fffffffde4c	0x00007fff
arr	0x7fffffffde48	0xffffde20
	0x7fffffffde44	
	0x7fffffffde40	
	0x7fffffffde3c	
	0x7fffffffde38	
	0x7fffffffde34	
	0x7fffffffde30	
	0x7fffffffde2c	0x00007fff
arr[1]	0x7fffffffde28	0xffffde3c
	0x7fffffffde24	0x00007fff
arr[0]	0x7fffffffde20	0xffffde30
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

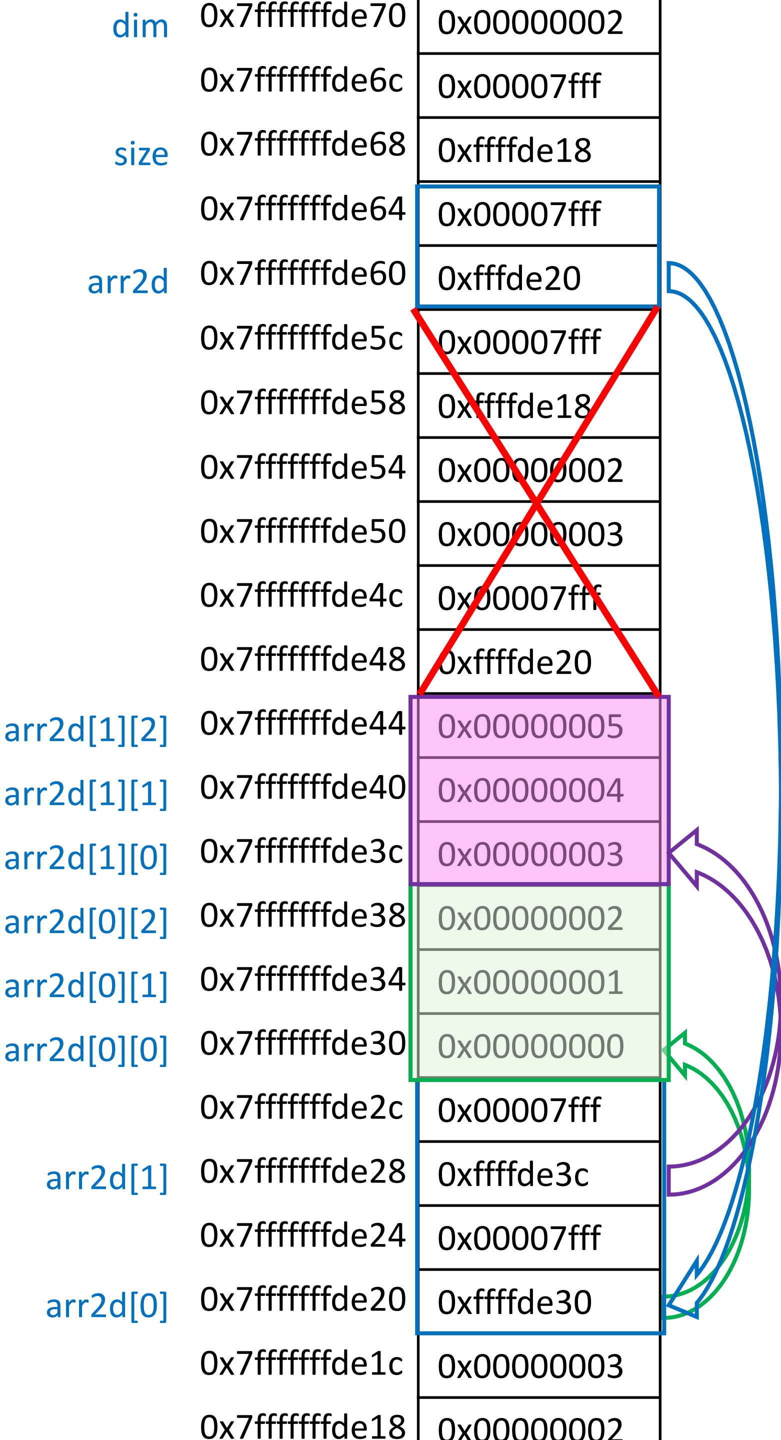
2차원 배열 만들어 반환하기 makeArray2D()



```cpp
#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 \
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
    for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
    cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

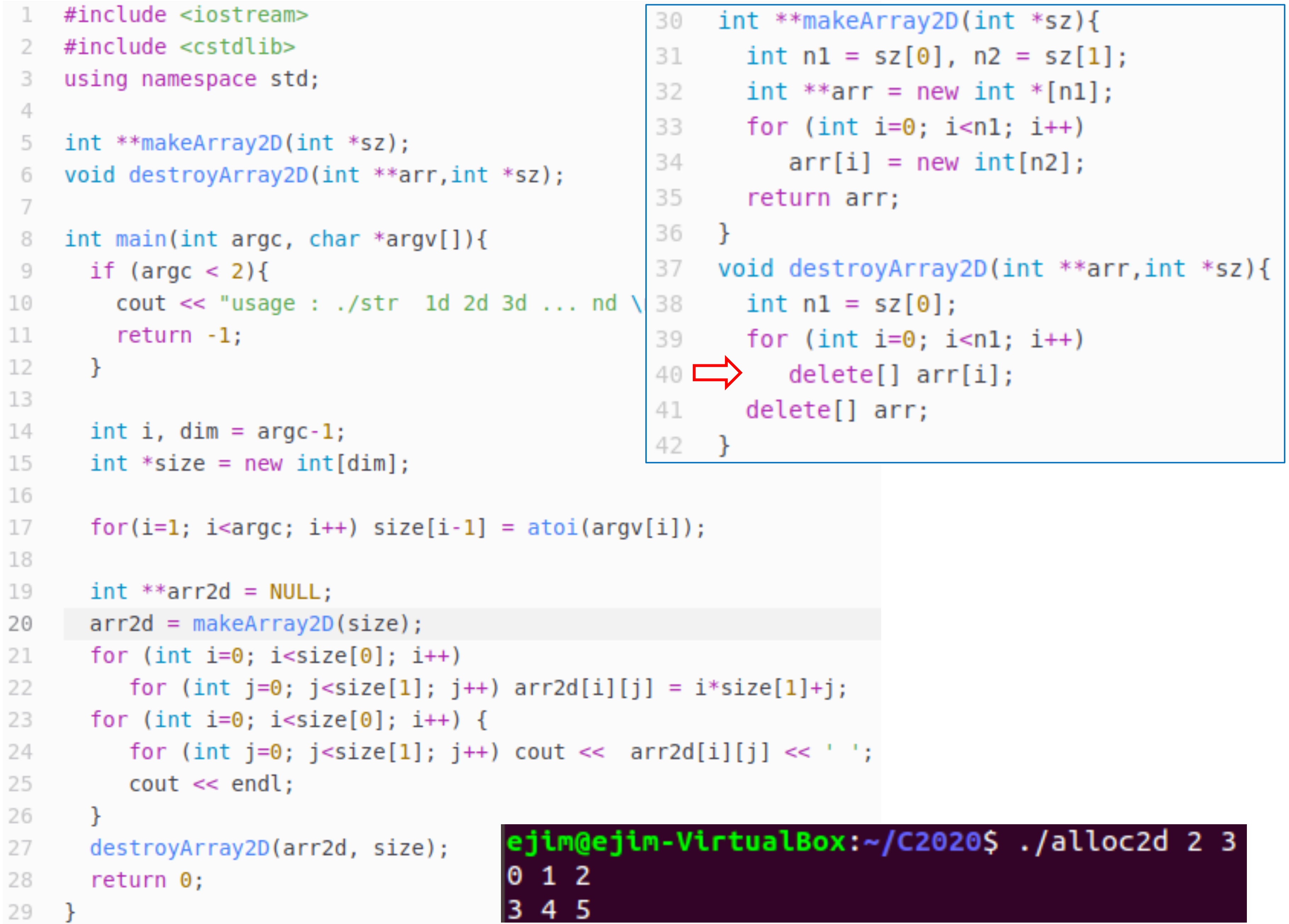
```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

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



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00007fff
arr2d	0x7fffffffde60	0xffffde20
	0x7fffffffde5c	0x00007fff
	0x7fffffffde58	0xffffde18
	0x7fffffffde54	0x00000002
	0x7fffffffde50	0x00000003
	0x7fffffffde4c	0x00007fff
	0x7fffffffde48	0xffffde20
arr2d[1][2]	0x7fffffffde44	0x00000005
arr2d[1][1]	0x7fffffffde40	0x00000004
arr2d[1][0]	0x7fffffffde3c	0x00000003
arr2d[0][2]	0x7fffffffde38	0x00000002
arr2d[0][1]	0x7fffffffde34	0x00000001
arr2d[0][0]	0x7fffffffde30	0x00000000
	0x7fffffffde2c	0x00007fff
arr2d[1]	0x7fffffffde28	0xffffde3c
	0x7fffffffde24	0x00007fff
arr2d[0]	0x7fffffffde20	0xffffde30
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

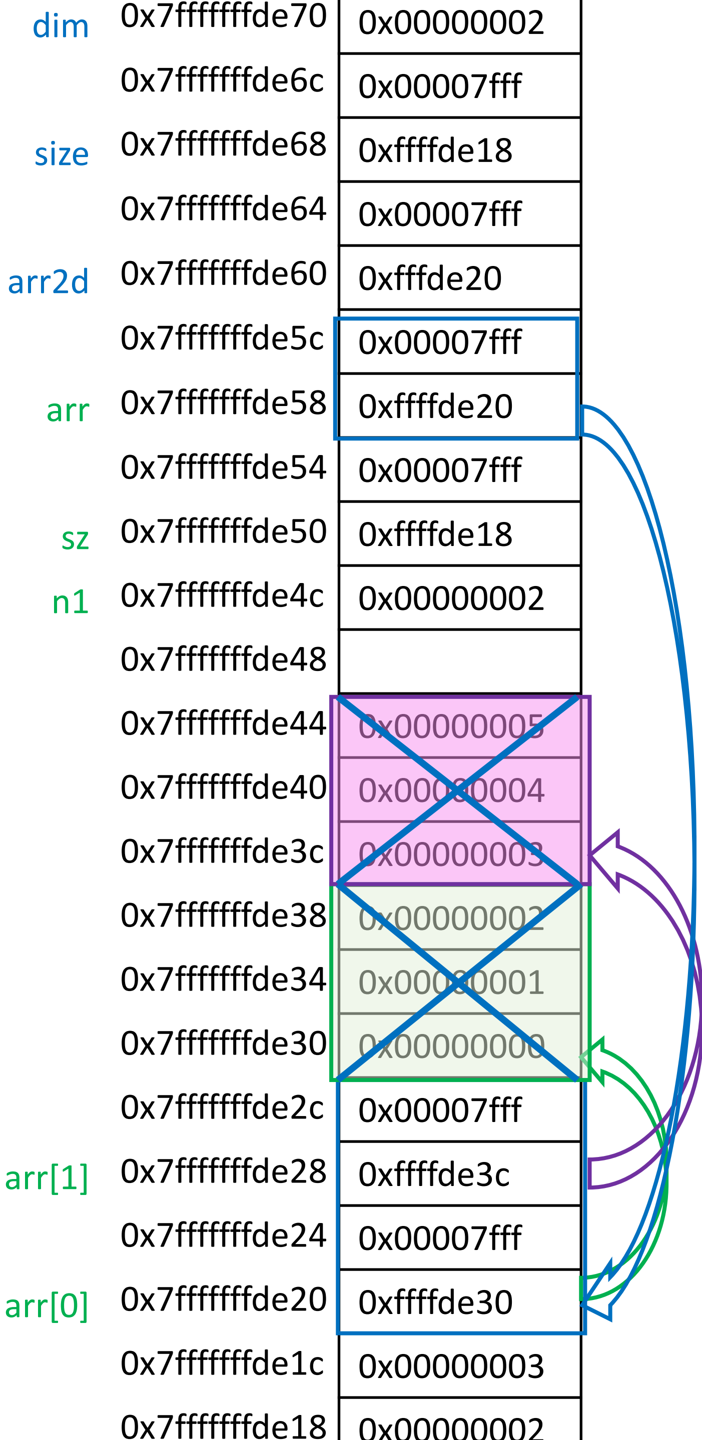
2차원 배열 destroyArray2D()



```cpp
#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 \
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

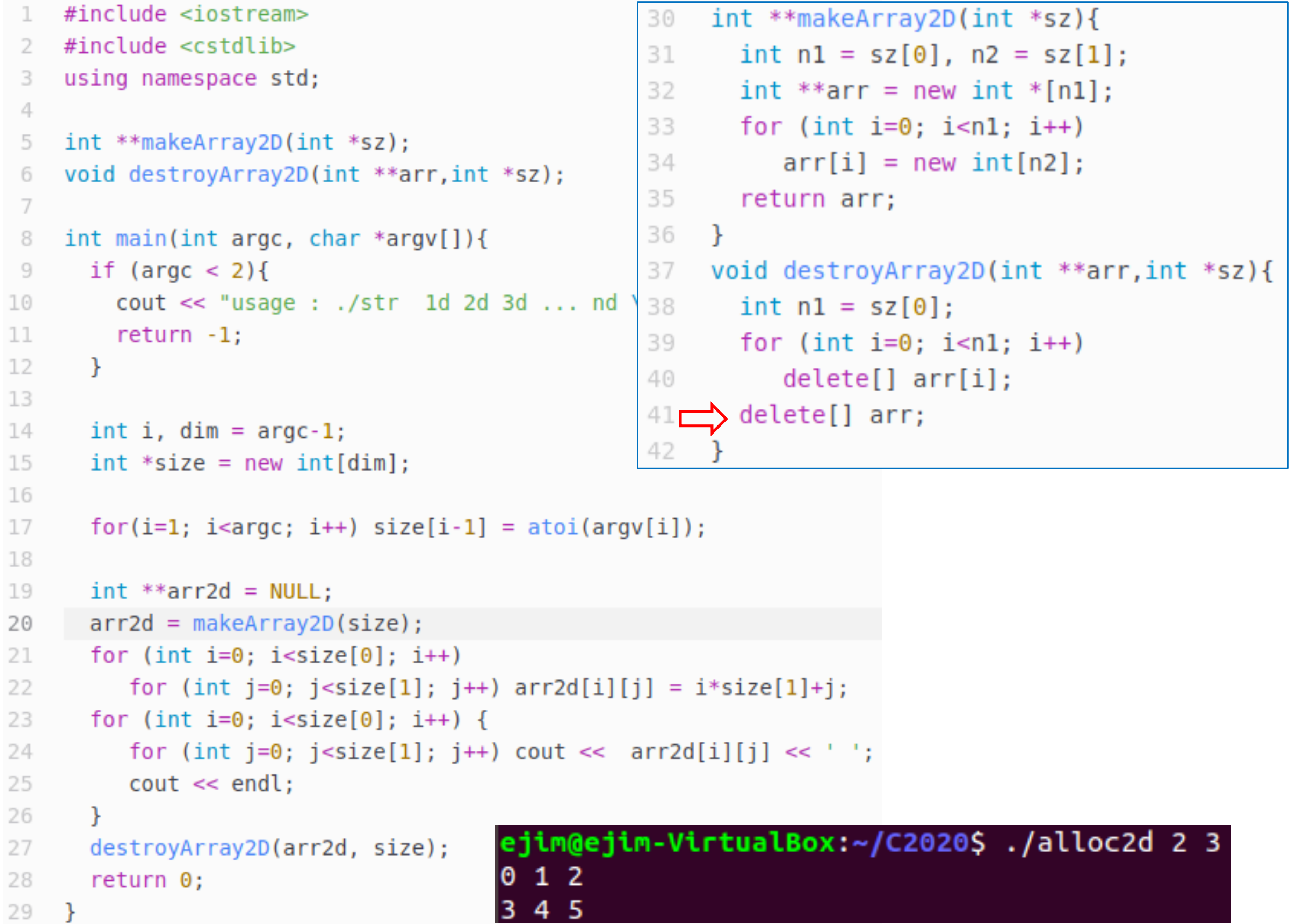
```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

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



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00007fff
arr2d	0x7fffffffde60	0xffffde20
	0x7fffffffde5c	0x00007fff
arr	0x7fffffffde58	0xffffde20
	0x7fffffffde54	0x00007fff
sz	0x7fffffffde50	0xffffde18
n1	0x7fffffffde4c	0x00000002
	0x7fffffffde48	
	0x7fffffffde44	~~0x00000005~~
	0x7fffffffde40	~~0x00000004~~
	0x7fffffffde3c	~~0x00000003~~
	0x7fffffffde38	~~0x00000002~~
	0x7fffffffde34	~~0x00000001~~
	0x7fffffffde30	~~0x00000000~~
	0x7fffffffde2c	0x00007fff
arr[1]	0x7fffffffde28	0xffffde3c
	0x7fffffffde24	0x00007fff
arr[0]	0x7fffffffde20	0xffffde30
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

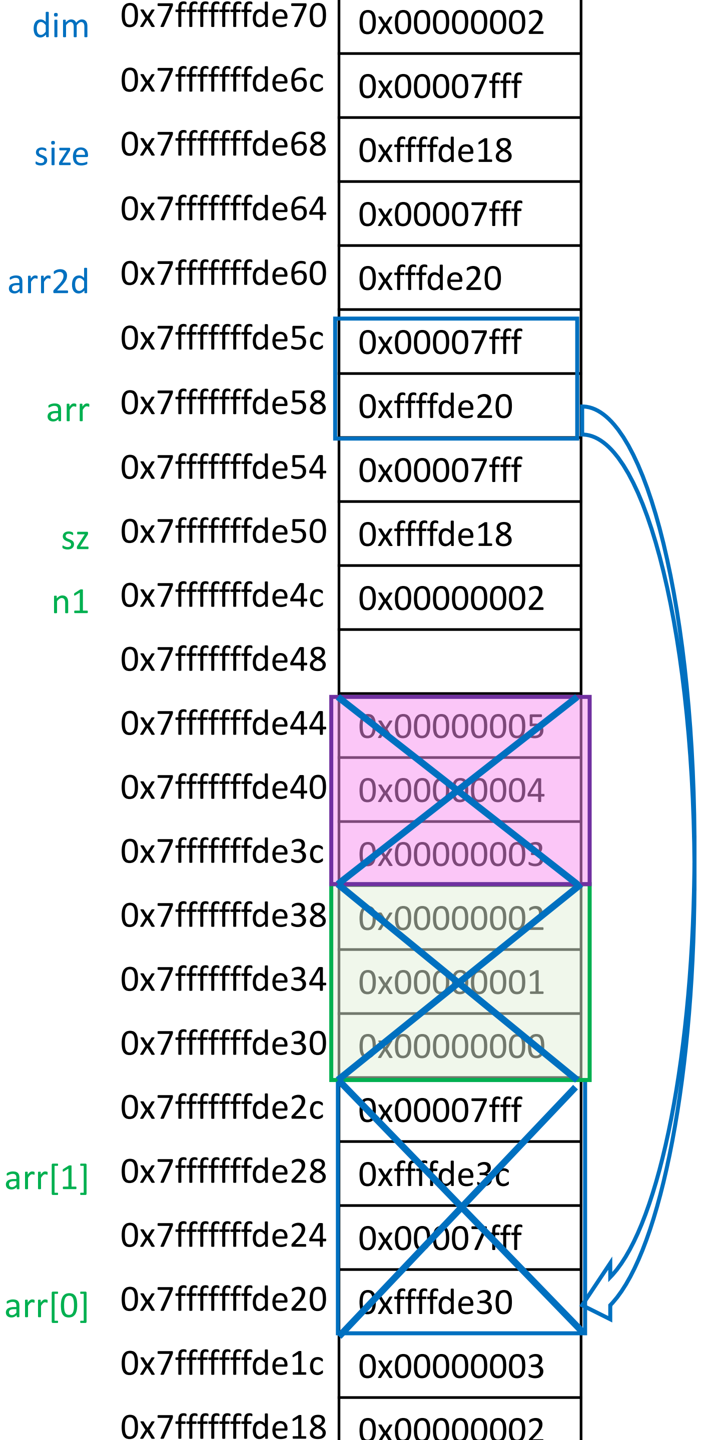
2차원 배열 destroyArray2D()



```cpp
#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 \
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

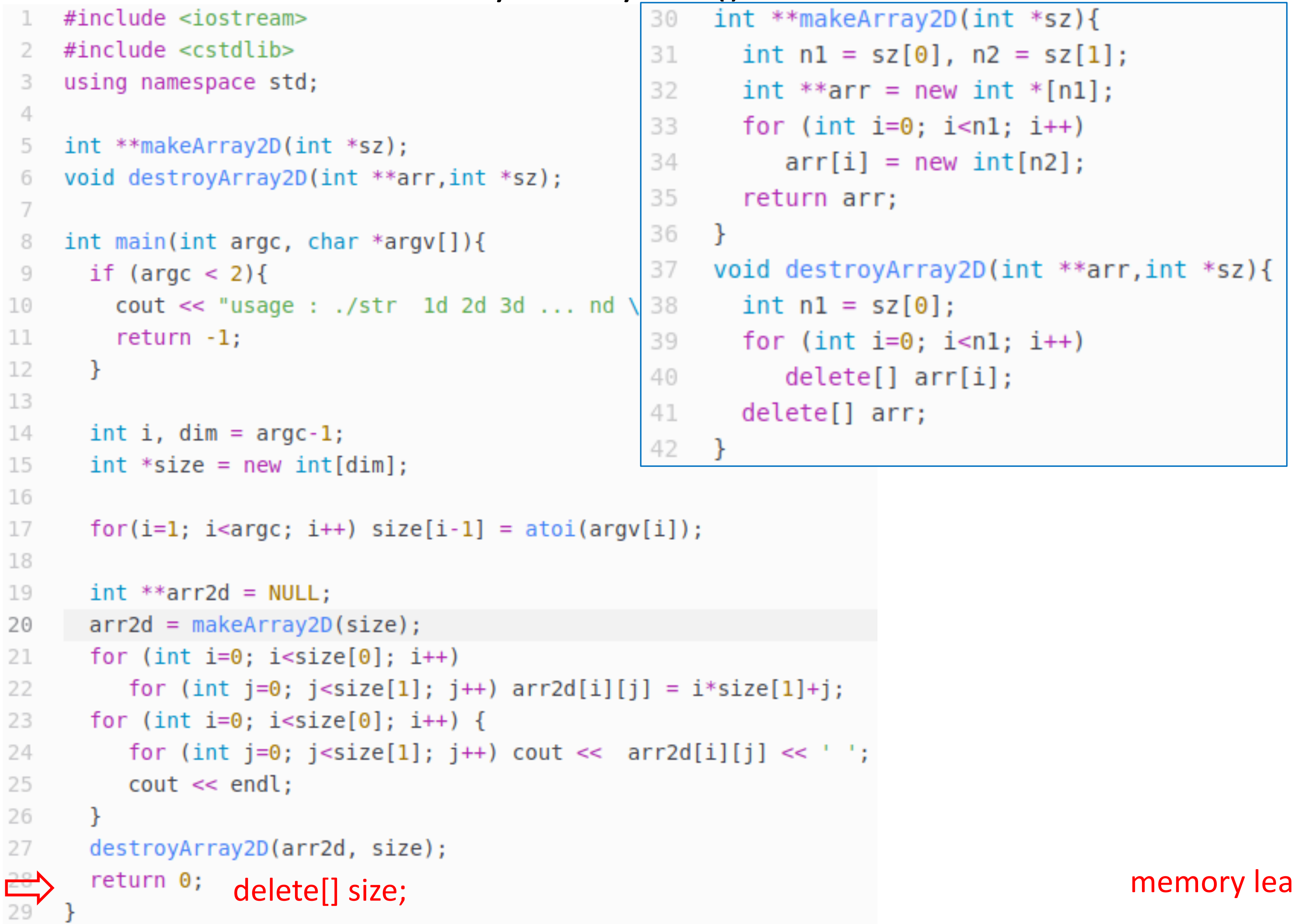
```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
⇨ delete[] arr;
}
```

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



dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00007fff
arr2d	0x7fffffffde60	0xffffde20
	0x7fffffffde5c	0x00007fff
arr	0x7fffffffde58	0xffffde20
	0x7fffffffde54	0x00007fff
sz	0x7fffffffde50	0xffffde18
n1	0x7fffffffde4c	0x00000002
	0x7fffffffde48	
	0x7fffffffde44	~~0x00000005~~
	0x7fffffffde40	~~0x00000004~~
	0x7fffffffde3c	~~0x00000003~~
	0x7fffffffde38	~~0x00000002~~
	0x7fffffffde34	~~0x00000001~~
	0x7fffffffde30	~~0x00000000~~
	0x7fffffffde2c	~~0x00007fff~~
arr[1]	0x7fffffffde28	~~0xffffde3c~~
	0x7fffffffde24	~~0x00007fff~~
arr[0]	0x7fffffffde20	~~0xffffde30~~
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

2차원 배열 destroyArray2D()

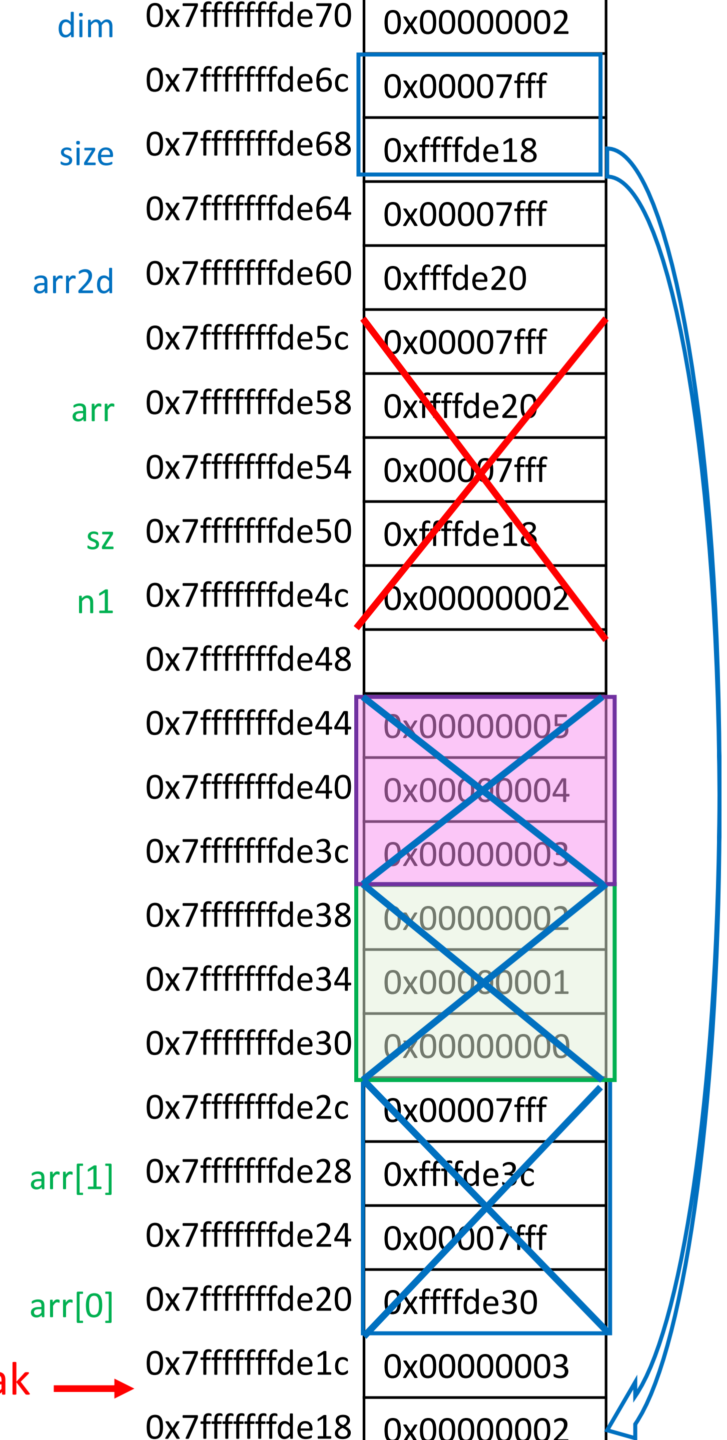


```cpp
#include <iostream>
#include <cstdlib>
using namespace std;

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 \
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int **arr2d = NULL;
  arr2d = makeArray2D(size);
  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++) {
     for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
     cout << endl;
  }
  destroyArray2D(arr2d, size);
  return 0;
}
```

```cpp
int **makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int **arr = new int *[n1];
  for (int i=0; i<n1; i++)
     arr[i] = new int[n2];
  return arr;
}
void destroyArray2D(int **arr,int *sz){
  int n1 = sz[0];
  for (int i=0; i<n1; i++)
     delete[] arr[i];
  delete[] arr;
}
```

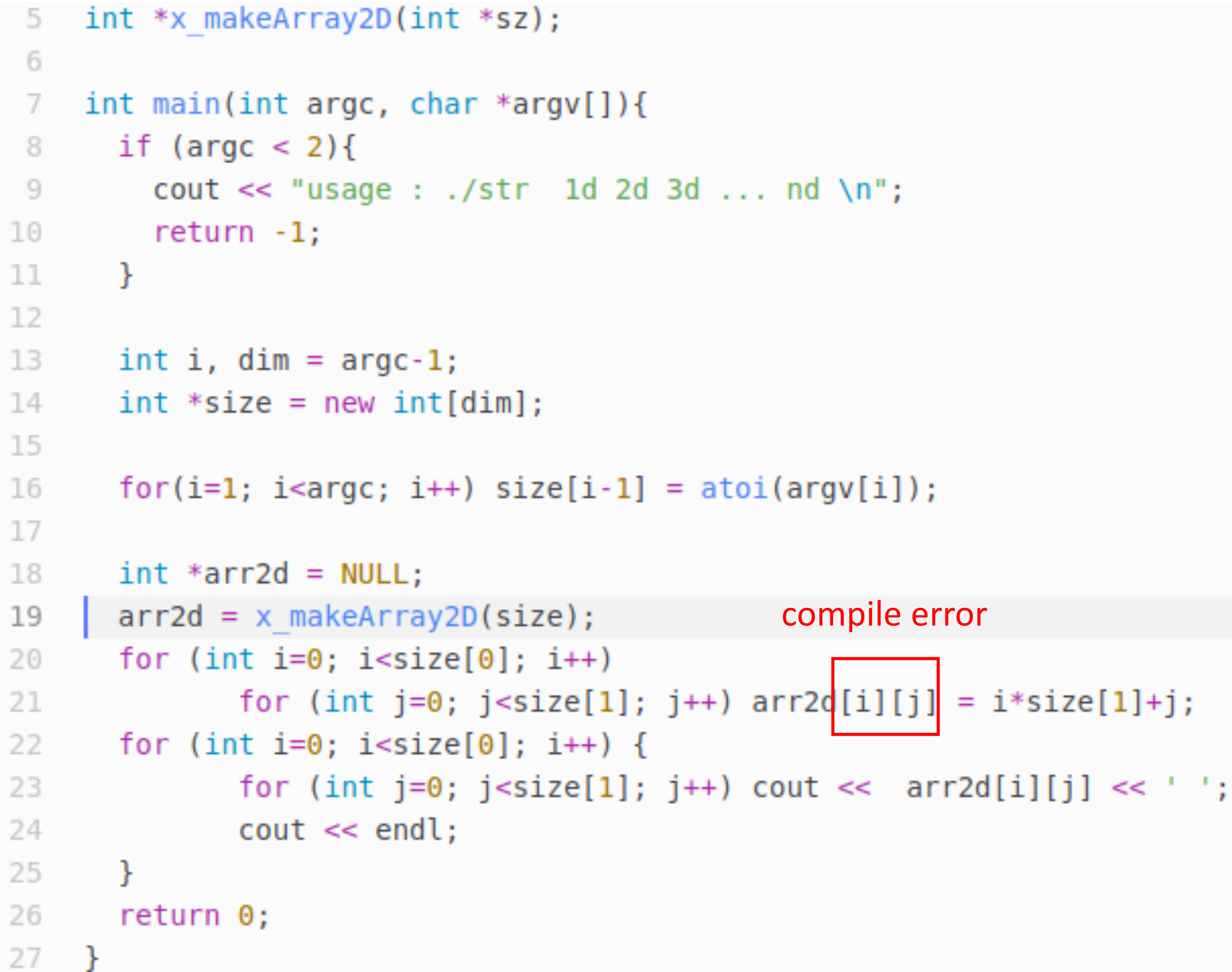
delete[] size;

memory leak



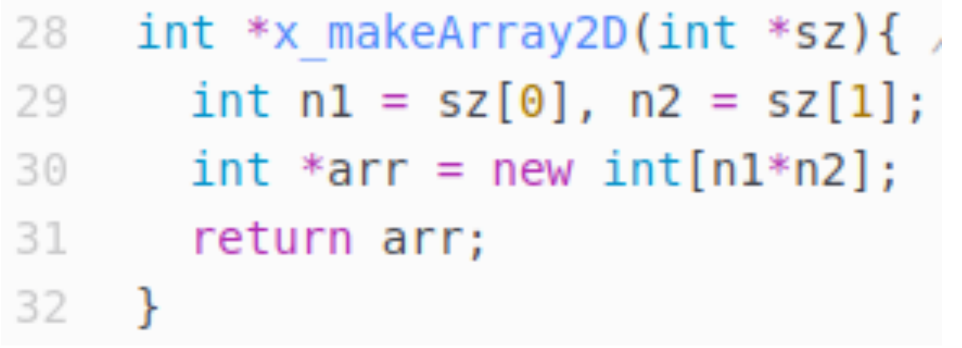
dim	0x7fffffffde70	0x00000002
	0x7fffffffde6c	0x00007fff
size	0x7fffffffde68	0xffffde18
	0x7fffffffde64	0x00007fff
arr2d	0x7fffffffde60	0xffffde20
	0x7fffffffde5c	0x00007fff
arr	0x7fffffffde58	0xffffde20
	0x7fffffffde54	0x00007fff
sz	0x7fffffffde50	0xffffde18
n1	0x7fffffffde4c	0x00000002
	0x7fffffffde48	
	0x7fffffffde44	0x00000005
	0x7fffffffde40	0x00000004
	0x7fffffffde3c	0x00000003
	0x7fffffffde38	0x00000002
	0x7fffffffde34	0x00000001
	0x7fffffffde30	0x00000000
	0x7fffffffde2c	0x00007fff
arr[1]	0x7fffffffde28	0xffffde3c
	0x7fffffffde24	0x00007fff
arr[0]	0x7fffffffde20	0xffffde30
	0x7fffffffde1c	0x00000003
	0x7fffffffde18	0x00000002

2차원 배열 : wrong example



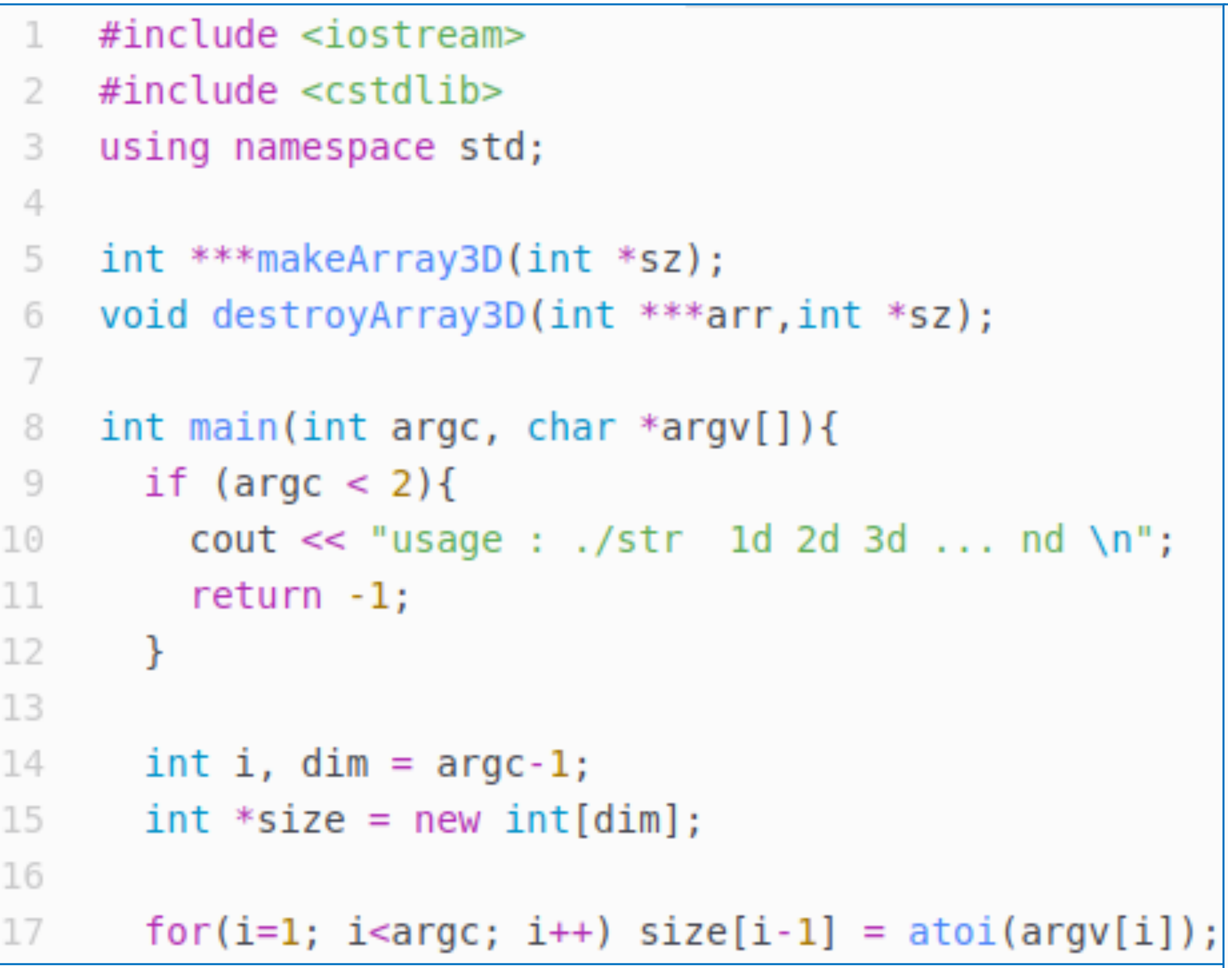
```
int *x_makeArray2D(int *sz);

int main(int argc, char *argv[]){
  if (argc < 2){
    cout << "usage : ./str  1d 2d 3d ... nd \n";
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);

  int *arr2d = NULL;
  arr2d = x_makeArray2D(size);
  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++) {
         for (int j=0; j<size[1]; j++) cout <<  arr2d[i][j] << ' ';
         cout << endl;
  }
  return 0;
}
```

compile error

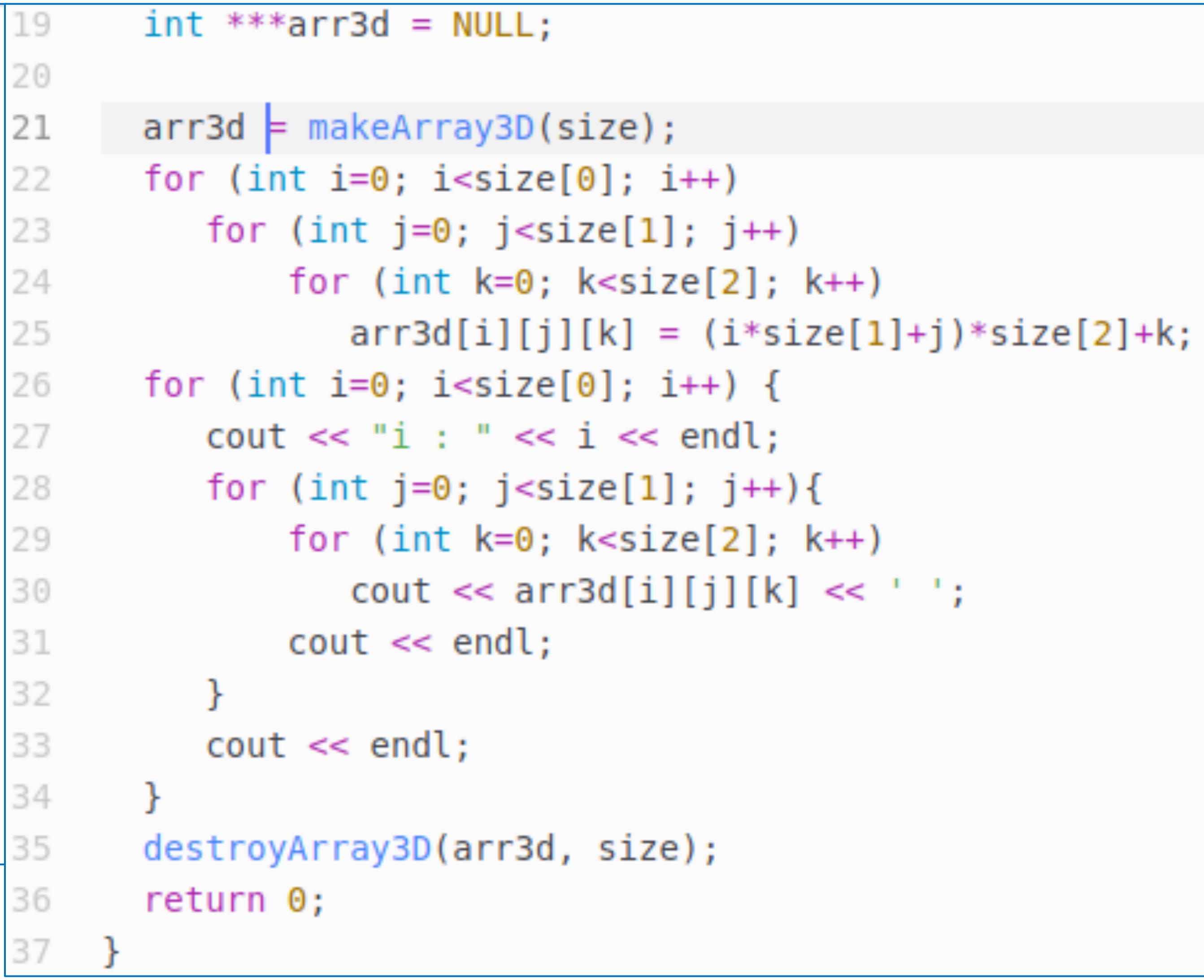


```
int *x_makeArray2D(int *sz){
  int n1 = sz[0], n2 = sz[1];
  int *arr = new int[n1*n2];
  return arr;
}
```

3차원 배열 만들어 반환하기 makeArray3D()

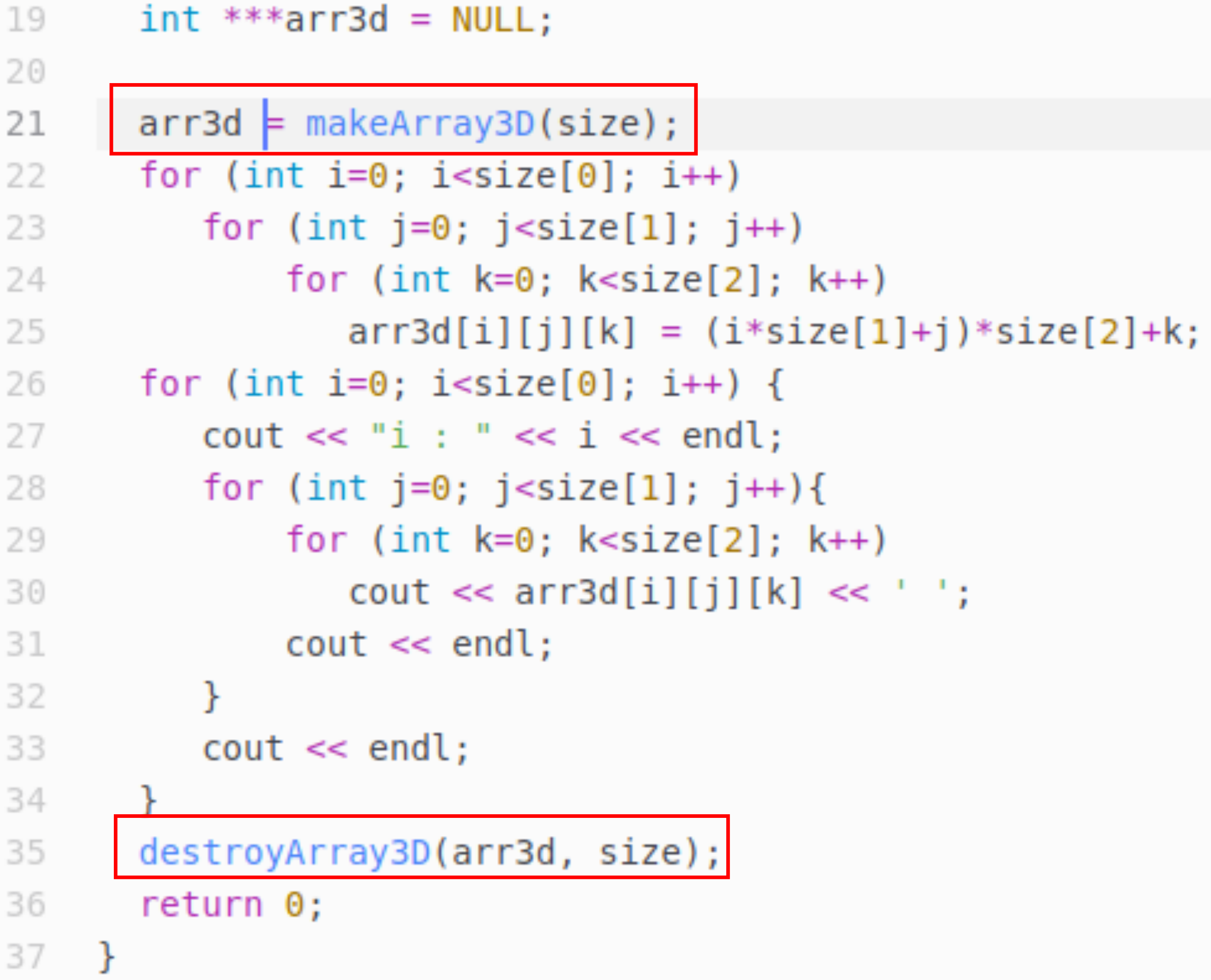


```cpp
#include <iostream>
#include <cstdlib>
using namespace std;

int ***makeArray3D(int *sz);
void destroyArray3D(int ***arr,int *sz);

int main(int argc, char *argv[]){
  if (argc < 2){
    cout << "usage : ./str  1d 2d 3d ... nd \n";
    return -1;
  }

  int i, dim = argc-1;
  int *size = new int[dim];

  for(i=1; i<argc; i++) size[i-1] = atoi(argv[i]);
```

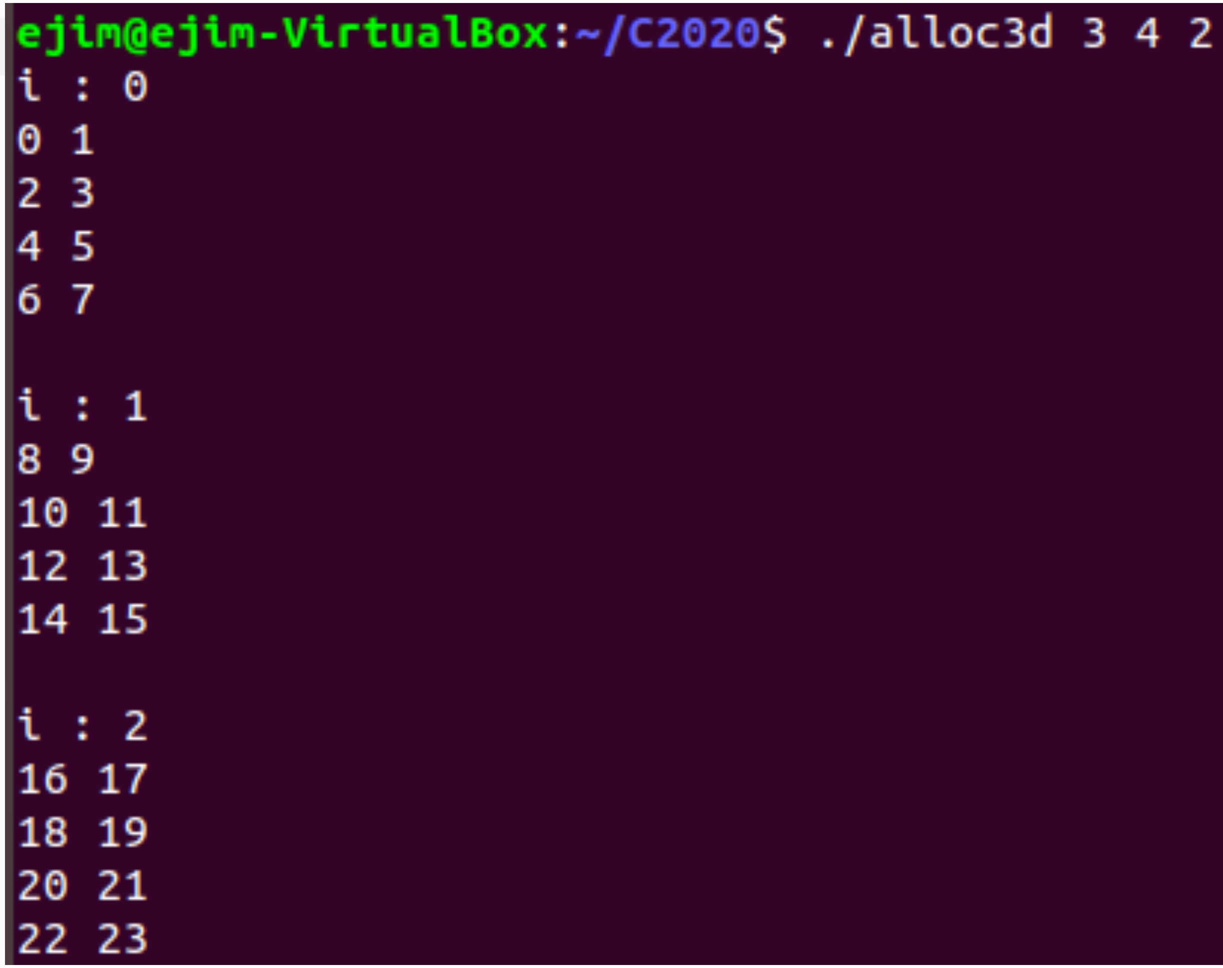


```cpp
  int ***arr3d = NULL;

  arr3d = makeArray3D(size);
  for (int i=0; i<size[0]; i++)
     for (int j=0; j<size[1]; j++)
         for (int k=0; k<size[2]; k++)
            arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
  for (int i=0; i<size[0]; i++) {
     cout << "i : " << i << endl;
     for (int j=0; j<size[1]; j++){
         for (int k=0; k<size[2]; k++)
            cout << arr3d[i][j][k] << ' ';
         cout << endl;
     }
     cout << endl;
  }
  destroyArray3D(arr3d, size);
  return 0;
}
```

3차원 배열 만들어 반환하기 makeArray3D()



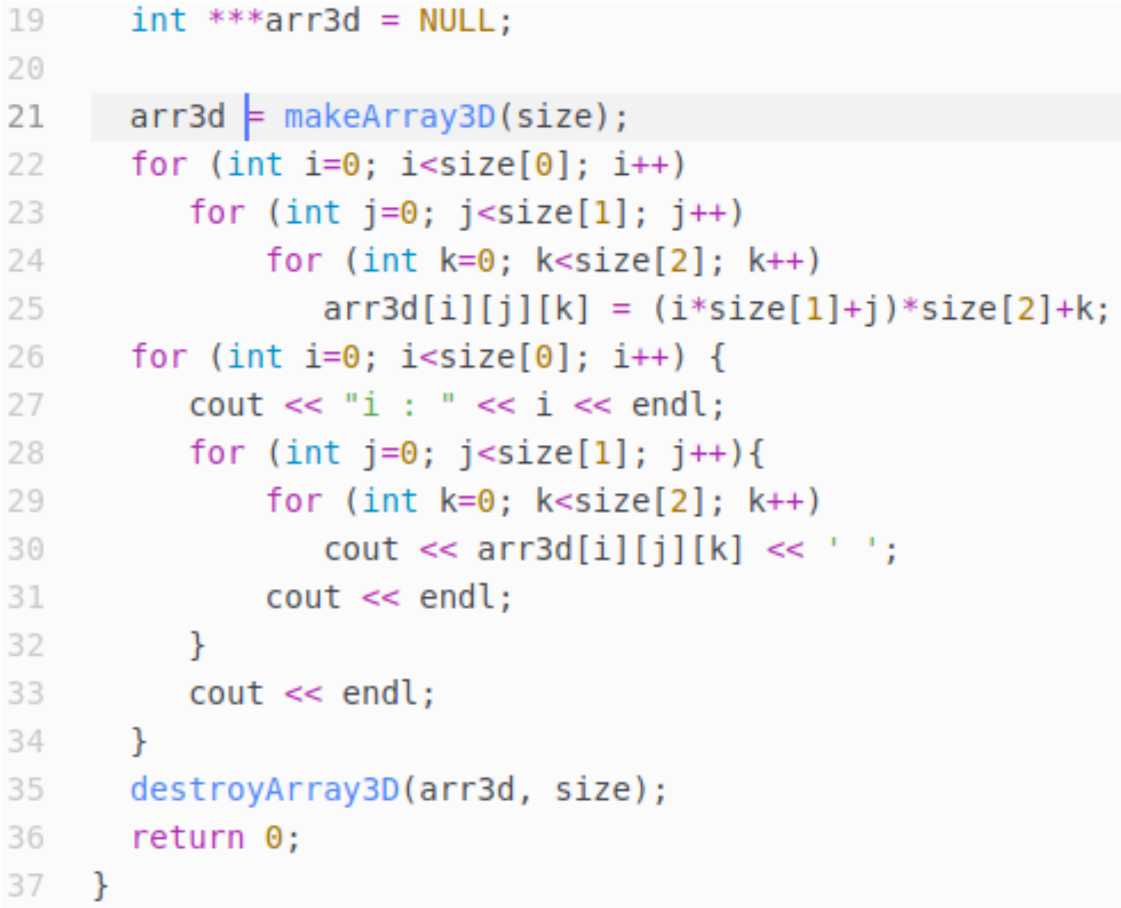
```
  int ***arr3d = NULL;

  arr3d = makeArray3D(size);
  for (int i=0; i<size[0]; i++)
     for (int j=0; j<size[1]; j++)
         for (int k=0; k<size[2]; k++)
            arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
  for (int i=0; i<size[0]; i++) {
     cout << "i : " << i << endl;
     for (int j=0; j<size[1]; j++){
         for (int k=0; k<size[2]; k++)
            cout << arr3d[i][j][k] << ' ';
         cout << endl;
     }
     cout << endl;
  }
  destroyArray3D(arr3d, size);
  return 0;
}
```



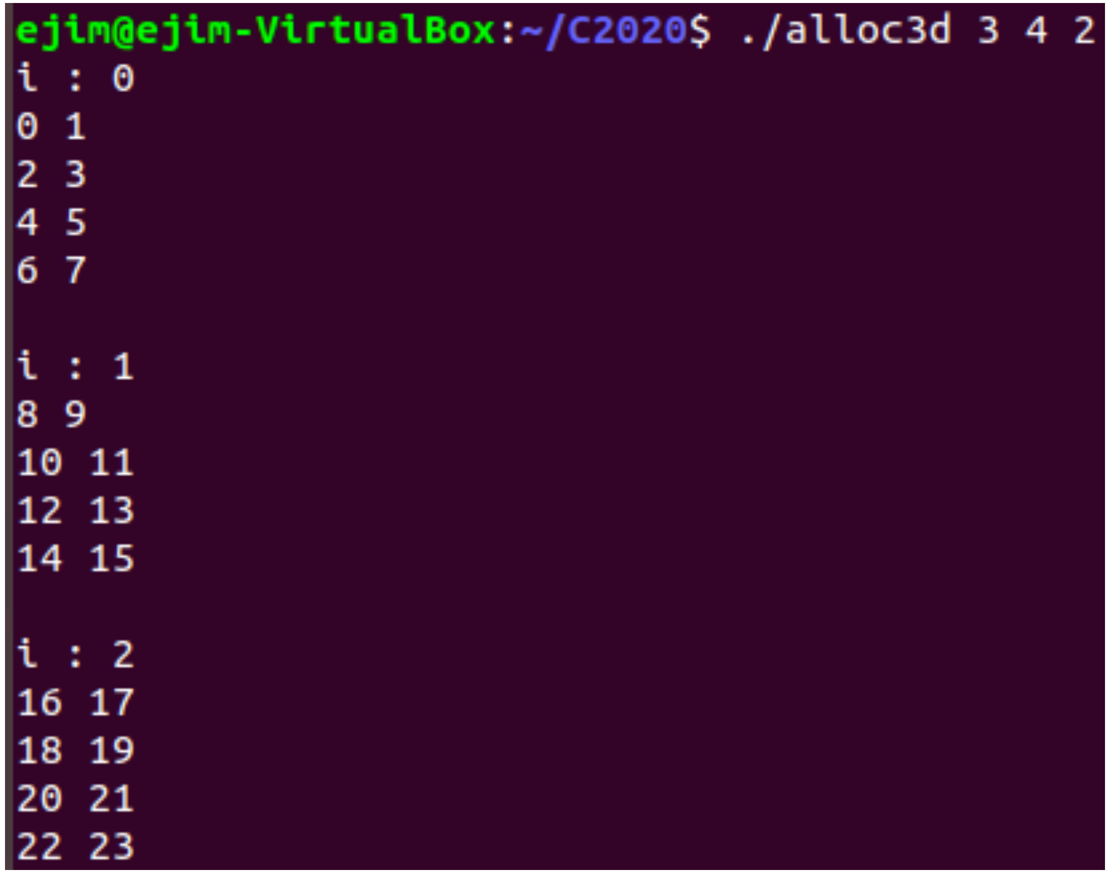
```
ejim@ejim-VirtualBox:~/C2020$ ./alloc3d 3 4 2
i : 0
0 1
2 3
4 5
6 7

i : 1
8 9
10 11
12 13
14 15

i : 2
16 17
18 19
20 21
22 23
```

실습

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



```
  int ***arr3d = NULL;

  arr3d = makeArray3D(size);
  for (int i=0; i<size[0]; i++)
     for (int j=0; j<size[1]; j++)
         for (int k=0; k<size[2]; k++)
            arr3d[i][j][k] = (i*size[1]+j)*size[2]+k;
  for (int i=0; i<size[0]; i++) {
     cout << "i : " << i << endl;
     for (int j=0; j<size[1]; j++){
         for (int k=0; k<size[2]; k++)
            cout << arr3d[i][j][k] << ' ';
         cout << endl;
     }
     cout << endl;
  }
  destroyArray3D(arr3d, size);
  return 0;
}
```



```
ejim@ejim-VirtualBox:~/C2020$ ./alloc3d 3 4 2
i : 0
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
```