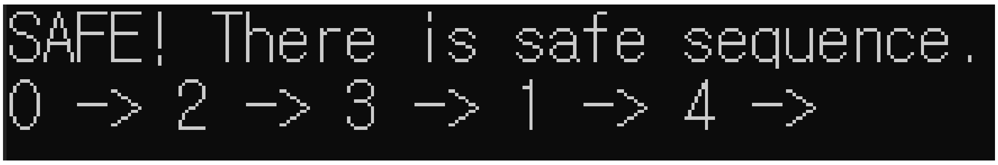
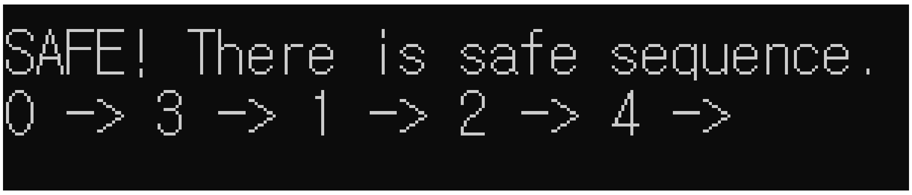
Bankers Algorithm 구현

1. Is the System is safe state? Yes.

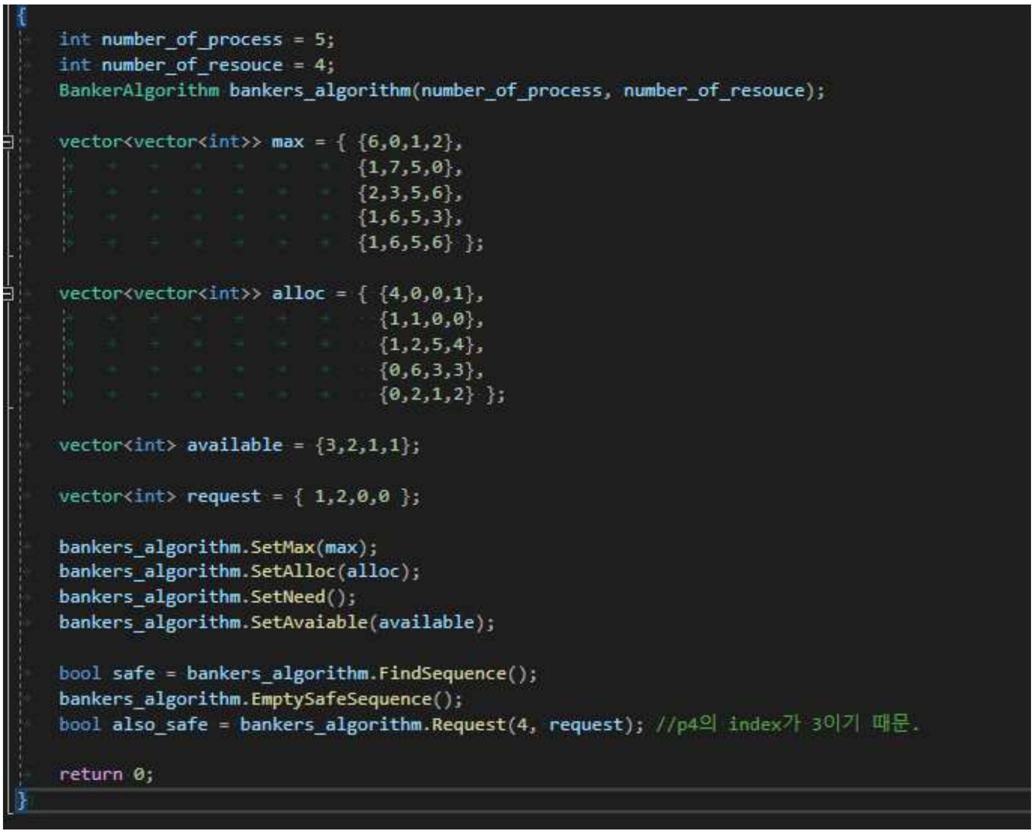


```
SAFE! There is safe sequence.
0 -> 2 -> 3 -> 1 -> 4 ->
```

2. Can request for (1,2,0,0) by P4 be granted? Yes



```
SAFE! There is safe sequence.
0 -> 3 -> 1 -> 2 -> 4 ->
```



```
{
    int number_of_process = 5;
    int number_of_resouce = 4;
    BankerAlgorithm bankers_algorithm(number_of_process, number_of_resouce);

    vector<vector<int>> max = { {6,0,1,2},
                                {1,7,5,0},
                                {2,3,5,6},
                                {1,6,5,3},
                                {1,6,5,6} };

    vector<vector<int>> alloc = { {4,0,0,1},
                                  {1,1,0,0},
                                  {1,2,5,4},
                                  {0,6,3,3},
                                  {0,2,1,2} };

    vector<int> available = {3,2,1,1};

    vector<int> request = { 1,2,0,0 };

    bankers_algorithm.SetMax(max);
    bankers_algorithm.SetAlloc(alloc);
    bankers_algorithm.SetNeed();
    bankers_algorithm.SetAvaiable(available);

    bool safe = bankers_algorithm.FindSequence();
    bankers_algorithm.EmptySafeSequence();
    bool also_safe = bankers_algorithm.Request(4, request); //p4의 index가 3이기 때문.

    return 0;
}
```