Easylab3—magic print

Magic Print —with print

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

```
int main() {
    cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

```
print content of result[63][61] 61
print content of result[63][62] 62
print content of result[63][63] 63
root@8a83d9ad86e1:~/easy_lab#
```

Magic Print—without print

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

```
int main() {
    // cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

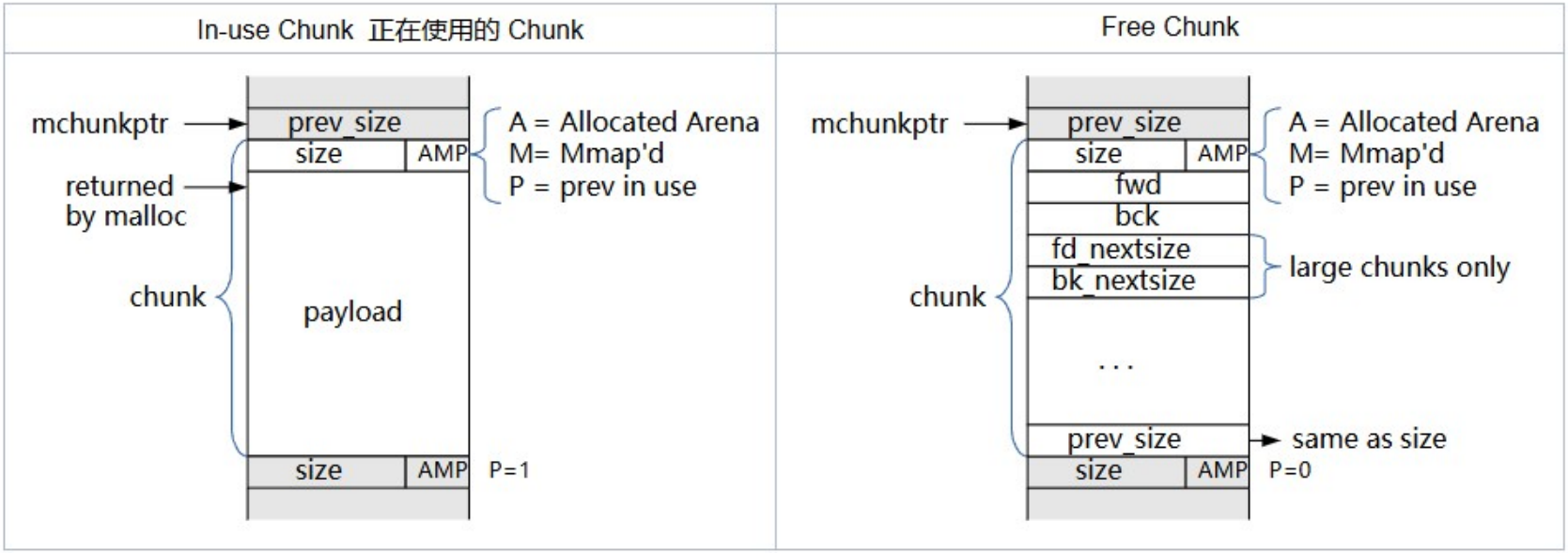
```
root@8a83d9ad86e1:~/easy_lab# g++ print.cpp -o print
root@8a83d9ad86e1:~/easy_lab# ./print
malloc(): corrupted top size
Aborted (core dumped)
root@8a83d9ad86e1:~/easy_lab#
```

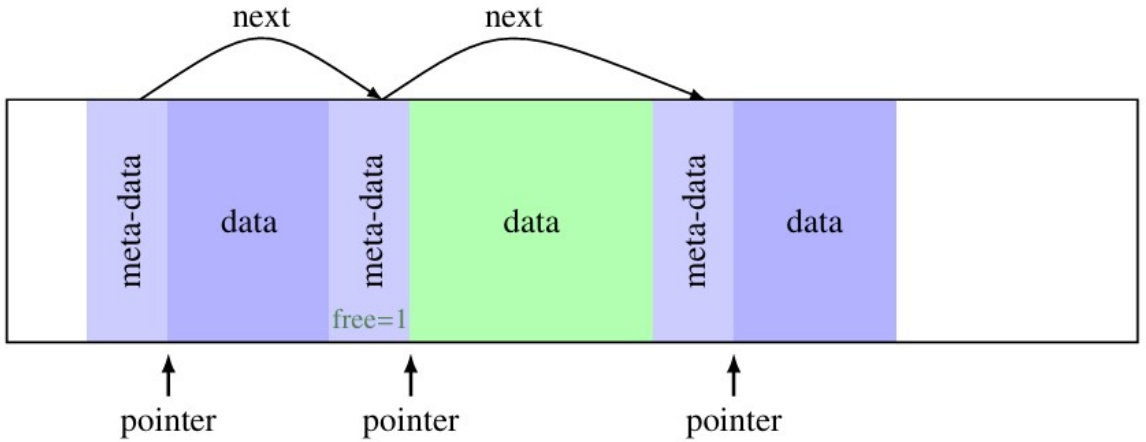
Magic Print—without print

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

```
int main() {
    // cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

meta-data in malloc



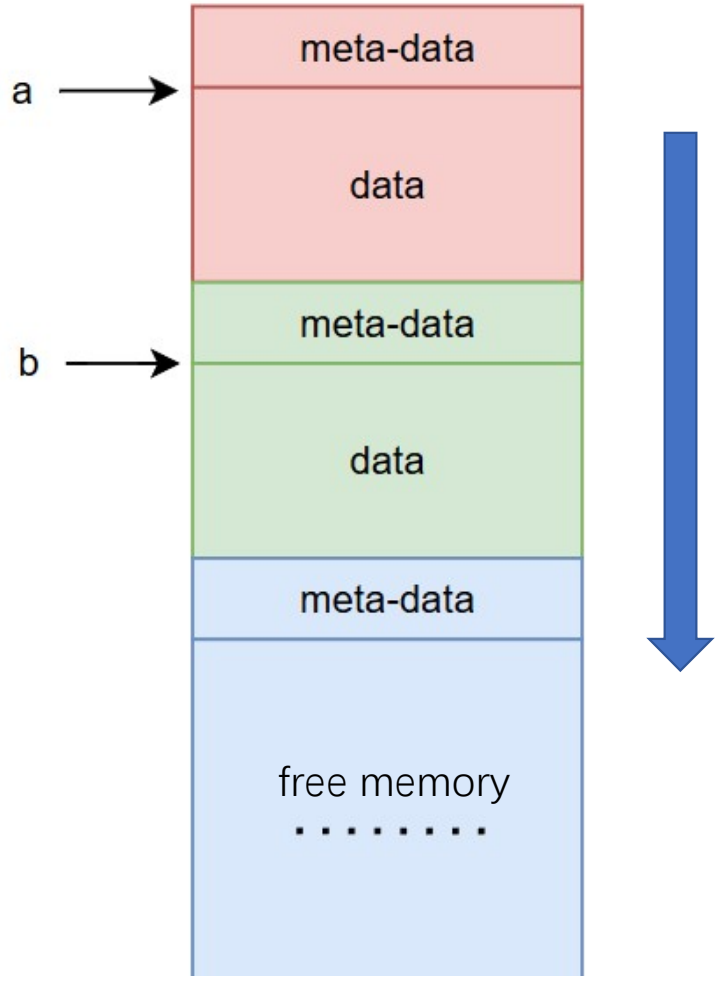


[1] https://sourceware.org/glibc/wiki/MallocInternals
[2] https://wiki-prog.infoprepa.epita.fr/images/0/04/Malloc_tutorial.pdf

meta-data in malloc

```
b.cpp
   5        {
   6            char *a = (char *)malloc(32);
   7            char *b = (char *)malloc(32);
B+>8            free(a);
   9            free(b);
   10       }
```

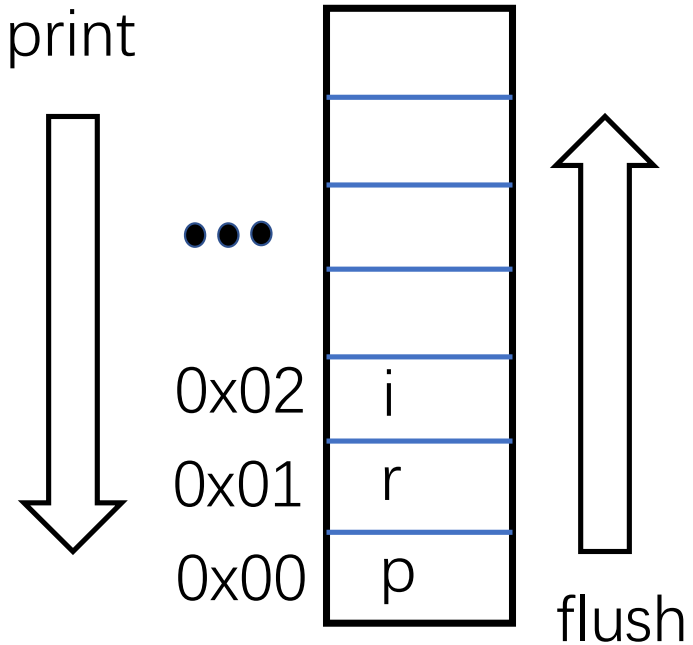
```
native process 4083843 In: main
(gdb) focus cmd
Focus set to cmd window.
(gdb) winheight cmd +20
(gdb) l
(gdb) b 8
Breakpoint 1 at 0x11d1: file b.cpp, line 8.
(gdb) r
Starting program: /root/easy_lab/b
warning: Error disabling address space randomization: Operation not permitted

Breakpoint 1, main () at b.cpp:8
(gdb) p b-a
$1 = 48
(gdb) x/112xb a-16
0x61dd9d9adea0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adea8: 0x31    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adeb0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adeb8: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adec0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adec8: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9aded0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9aded8: 0x31    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adee0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adee8: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adef0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adef8: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adf00: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x61dd9d9adf08: 0x01    0xf1    0x00    0x00    0x00    0x00    0x00    0x00
(gdb)
```



How does `Print` affiliated with memory model

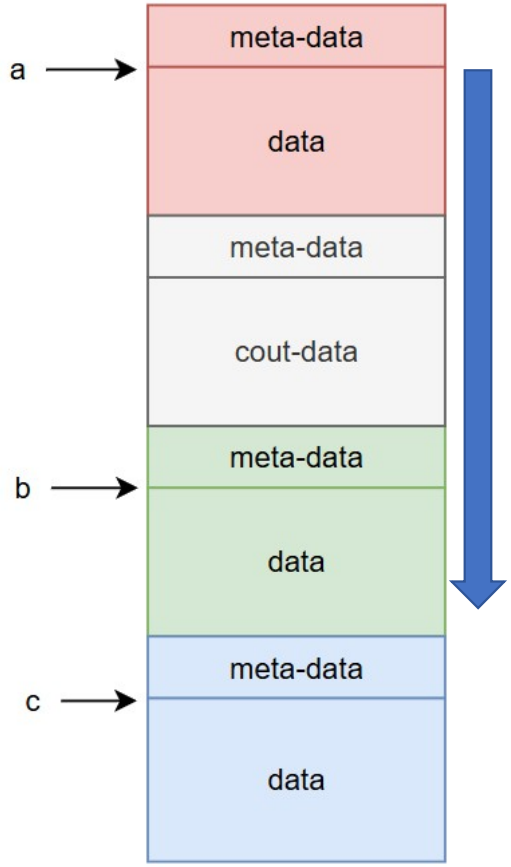
```
cout<<"A magic print! If you comment this, the program will break."<<endl;
```

Print mallocs buffer, and flushes the buffer




```
cout<<"print buffer" <<endl;
```

```
b.cpp
5           {
6               char *a = (char *)malloc(32);
7               cout<<"malloc a"<<endl;
8               char *b = (char *)malloc(32);
9               cout<<"malloc b"<<endl;
10              char *c = (char *)malloc(32);
B+>11           free(a);
12              free(b);
13              free(c);
14          }

native process 4098562 In: main
(gdb) focus cmd
Focus set to cmd window.
(gdb) winheight cmd +20
(gdb) l
(gdb) b 11
Breakpoint 1 at 0x126f: file b.cpp, line 11.
(gdb) r
Starting program: /root/easy_lab/b
warning: Error disabling address space randomization: Operation not permitted
malloc a
Breakpoint 1, main () at b.cpp:11
(gdb) p a
$1 = 0x5e376093deb0 ""
(gdb) p b
$2 = 0x5e376093e2f0 ""
(gdb) p c
$3 = 0x5e376093e320 ""
(gdb) p b-a
$4 = 1088
(gdb) p c-b
$5 = 48
(gdb)
```

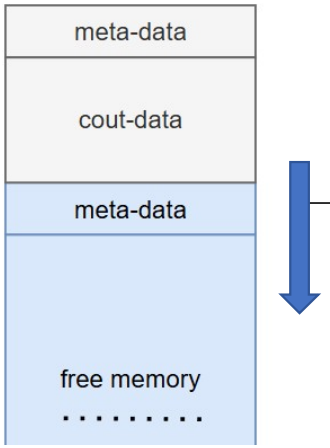


Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

1

```
int main() {
    cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```



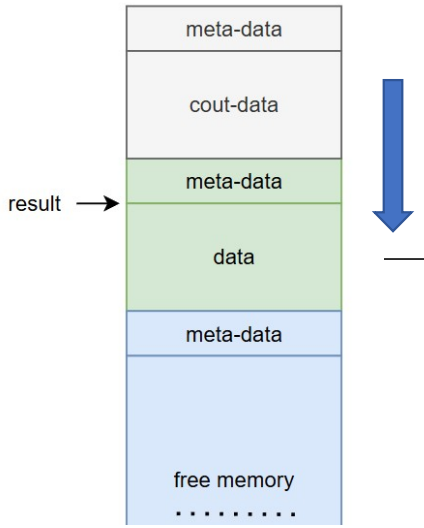
Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

2

```
int main() {
    cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

1



Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

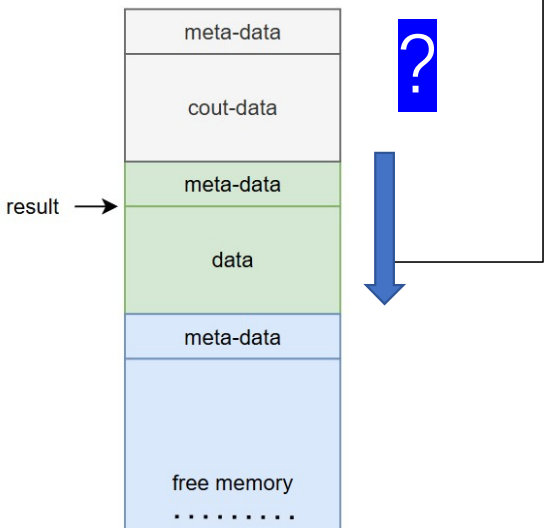
2

```
int main() {
    cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

1

?

?



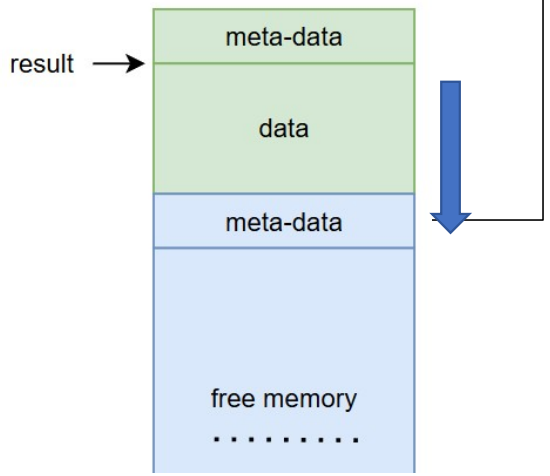
Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

1

no

```
int main() {
    // cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```



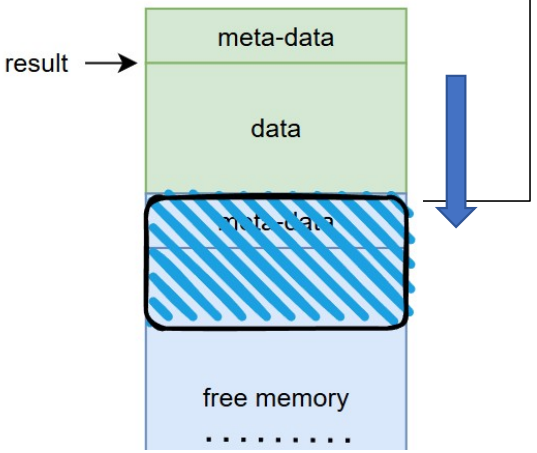
Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

1

no

```
int main() {
    // cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```



Program analysis

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

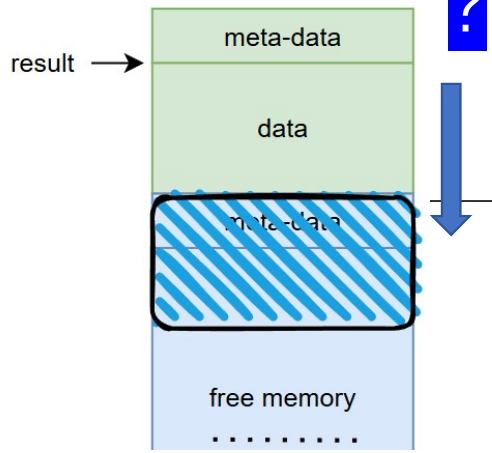
1

no

```
int main() {
    // cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

2

?



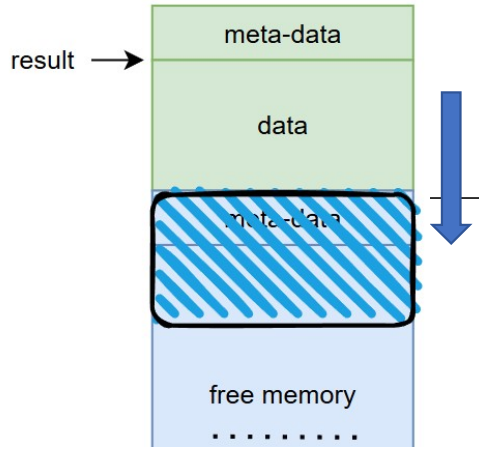
What's the problem?

```
root@8a83d9ad86e1:~/easy_lab# g++ print.cpp -o print
root@8a83d9ad86e1:~/easy_lab# ./print
malloc(): corrupted top size
Aborted (core dumped)
root@8a83d9ad86e1:~/easy_lab#
```

```
print.cpp
  16        int main() {
  17            // cout<<"A magic print! If you comment this, the program will break."<<endl;
  18            int **result = double_array(array_number);
  19
  20            for (int i = 0; i < array_number; ++i) {
B+>21                cout<<"print address of result["<<i<<"] "<<&result[i][0]<<endl;
  22                for (int j = 0; j < array_number; j++) {
  23                    result[i][j] = j;
  24                    cout<<"print content of result["<<i<<"]["<<j<<"] "<<result[i][j]<<endl;
  25                }
  26            }
  27            free(result);
  28        }
```

```
native process 4110120 In: main
(gdb) fucos cmd
Undefined command: "fucos".  Try "help".
(gdb) focus cmd
Focus set to cmd window.
(gdb) winheight cmd +20
(gdb) l
(gdb) b 21
Breakpoint 1 at 0x12ff: file print.cpp, line 21.
(gdb) r
Starting program: /root/easy_lab/print
warning: Error disabling address space randomization: Operation not permitted

Breakpoint 1, main () at print.cpp:21
(gdb) x/16xb a+64
Argument to arithmetic operation not a number or boolean.
(gdb) x/16xb (char *)a+64
0x3e4bb67a00000040:     Cannot access memory at address 0x3e4bb67a00000040
(gdb) x/16xb result +8
0x60a79acf8ef0: 0x60    0x59    0xfb    0x99    0xa7    0x60    0x00    0x00
0x60a79acf8ef8: 0x60    0x5a    0xfb    0x99    0xa7    0x60    0x00    0x00
(gdb) p &matrix[8][0]
$1 = (int *) 0x60a799fb5960 <matrix+2048>
(gdb) p &matrix[9][0]
$2 = (int *) 0x60a799fb5a60 <matrix+2304>
(gdb)
```

```
int main() {
    // cout<<"A magic print! If you comment this, the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i] "<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j] "<<result[i][j]<<endl;
        }
    }
    free(result);
}
```



Think time

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

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            if(i == 8 || i == 9) continue;
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

```
int main() {
    // cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

What if we don’t edit the meta-data?

Think time

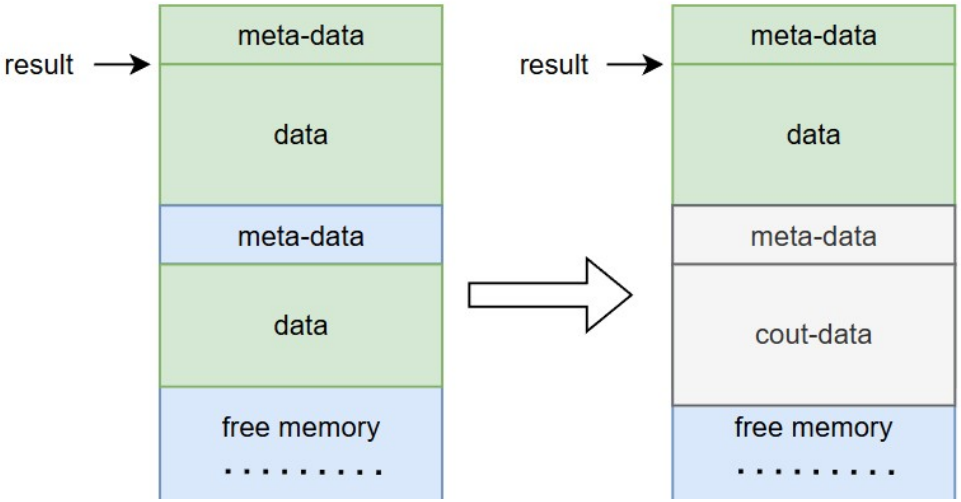
```
#include <iostream>
using namespace std;

#define array_number 64

int matrix[array_number][array_number];

int **double_array(size_t n) {
    int **result = new int*[8];

        for (int i = 0; i < n; ++i) {
            if(i == 8 || i == 9) continue;
            result[i] = matrix[i];
            for (int j = 0; j < n; ++j){
                result[i][j] = j;
            }
    }

    return result;
}
```

```
int main() {
    // cout<<"A magic print! If you comment this,
the program will break."<<endl;
    int **result = double_array(array_number);

    for (int i = 0; i < array_number; ++i) {
        cout<<"print address of result[i]
"<<&result[i][0]<<endl;
        for (int j = 0; j < array_number; j++) {
            result[i][j] = j;
            cout<<"print content of result[i][j]
"<<result[i][j]<<endl;
        }
    }
    free(result);
}
```

```
print content of result[7][62] 62
print content of result[7][63] 63
print address of result[8] 0
Segmentation fault (core dumped)
root@8a83d9ad86e1:~/easy_lab#
```

Think time

```
print content of result[7][62] 62
print content of result[7][63] 63
print address of result[8] 0
Segmentation fault (core dumped)
root@8a83d9ad86e1:~/easy_lab#
```



print.cpp

```
16          int main() {
17              // cout<<"A magic print! If you comment this, the program will break."<<endl;
18              int **result = double_array(array_number);
19
20              for (int i = 0; i < array_number; ++i) {
B+>21                   cout<<"print address of result["<<i<<"] "<<&result[i][0]<<endl;
22                  for (int j = 0; j < array_number; j++) {
23                      result[i][j] = j;
24                      cout<<"print content of result["<<i<<"]["<<j<<"] "<<result[i][j]<<endl;
25                  }
26              }
27              free(result);
28          }
```

```
native process 13945 In: main
(gdb) focus cmd
Focus set to cmd window.
(gdb) winheight cmd +20
(gdb) l
(gdb) b 21 if i>=8
Breakpoint 1 at 0x1313: file print.cpp, line 21.
(gdb) r
Starting program: /root/easy_lab/print
warning: Error disabling address space randomization: Operation not permitted

Breakpoint 1, main () at print.cpp:21
(gdb) p i
$1 = 8
(gdb) x/16xb result+8
0x59fdedac9ef0: 0x00    0x00    0x00    0x00    0x00    0x00    0x00    0x00
0x59fdedac9ef8: 0x11    0x04    0x00    0x00    0x00    0x00    0x00    0x00
(gdb) x/s result+10
0x59fdedac9f00: "print content of result[7][63] 63\nd4a5860\nJ\355\375Y"
(gdb) x/16xb result+10
0x59fdedac9f00: 0x70    0x72    0x69    0x6e    0x74    0x20    0x63    0x6f
0x59fdedac9f08: 0x6e    0x74    0x65    0x6e    0x74    0x20    0x6f    0x66
(gdb) p result[7]
$2 = (int *) 0x59fded4a5860 <matrix+1792>
(gdb)
```