[System Programming] Assignment #4

Spring 2025

Seulki Lee
Yunseok Lee
Kyu Hwan Lee
Jiwoon Chang
Yeojin Lee

CONTACT
Ulsan National Institute of Science and Technology
Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

Computer Science and Engineering
106 3rd Engineering Bldg
Tel. +82 52 217 6333 **Web.** https://cse.unist.ac.kr/

Assignment #4 (100 points)

- Due May 30, 2025: 11:59pm

- Platform
  - We will work on Ubuntu 22.10 (latest version)
  - https://releases.ubuntu.com/kinetic/ (Desktop image)

- If you use MAC, please use Docker Desktop on Mac
  - https://docs.docker.com/desktop/install/mac-install/

- If you cannot make this environment, please contact our TA
  - Yunseok Lee: walk1009@unist.ac.kr
  - Kyu Hwan Lee: hanbitchan@unist.ac.kr
  - Jiwoon Chang: jwc9876@unist.ac.kr
  - Yeojin Lee: yeojin@unist.ac.kr

Goal

- Write a custom allocator that manages a heap area minimally
- Your allocator manages 128 bytes of a heap area

What you need to implement

A `program` that

- allocates **128** bytes of a heap area and manages it (suppose that there is not enough memory you can use)
- runs in *an infinite while loop* where it gets input from users through ***stdin***
- *runs in a **64-bit** environment and follows the little-endian format*

Users can request a type of data (which will be allocated in the heap area) and its value to the program, then the program *allocates a room* in the heap area and assigns the value to the room

- e.g., a user can request a struct which has 2 int types and specify values for each int type in the struct

What you need to implement

There must be no padding bytes around any data in the heap area
Also, there must be no padding bytes in any struct type data

Also, users can request *deallocation* of any data

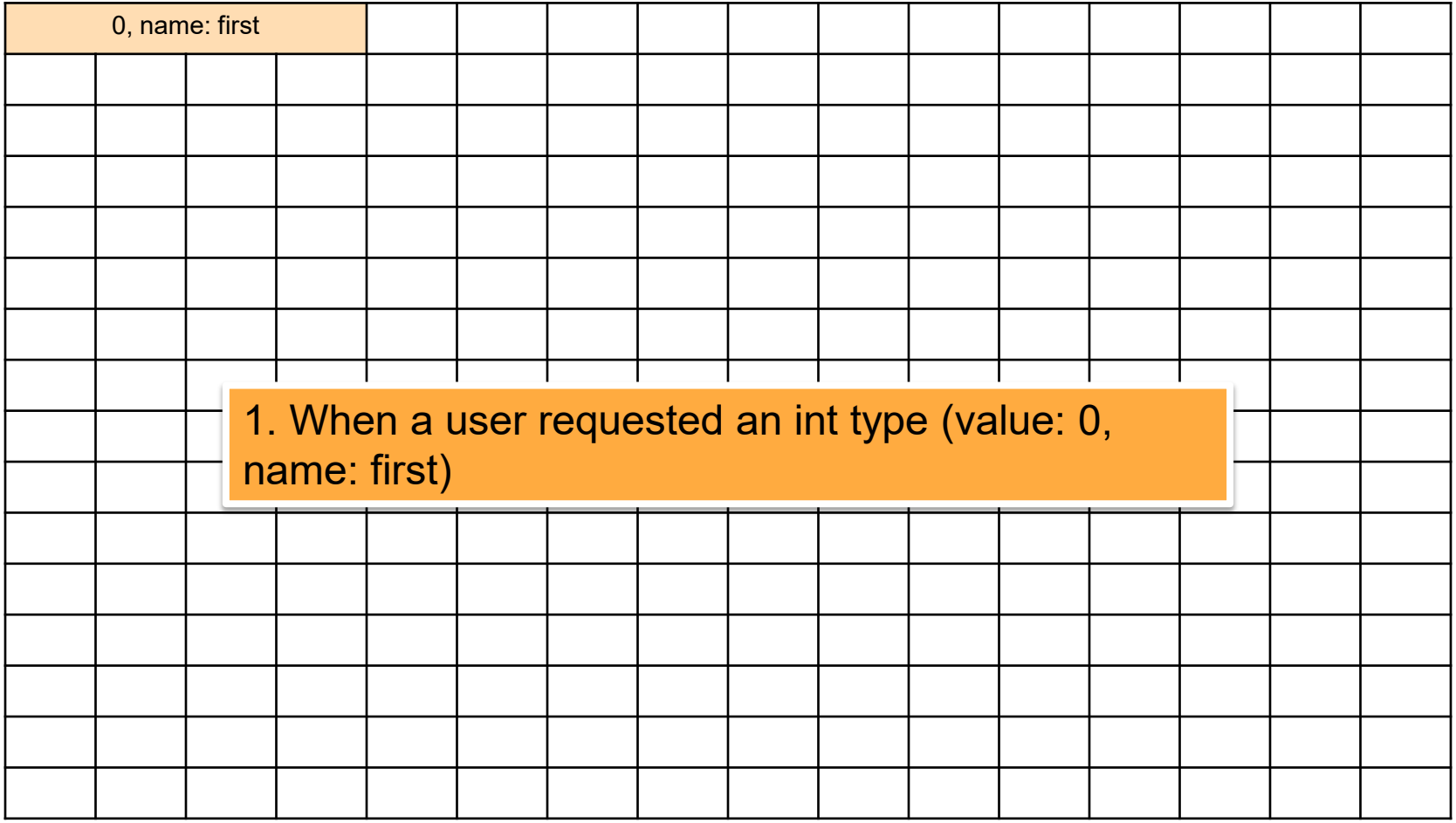
- When a deallocation of data requested, your program must move the other data to fill the hole (unless the data to be deallocated is stored at the last part of the heap area)

Your program must print out memory dump of the heap area and data list (see printing format section)

Example

1. When a user requested an int type (value: 0, name: first)

Example


0, name: first
1. When a user requested an int type (value: 0, name: first)

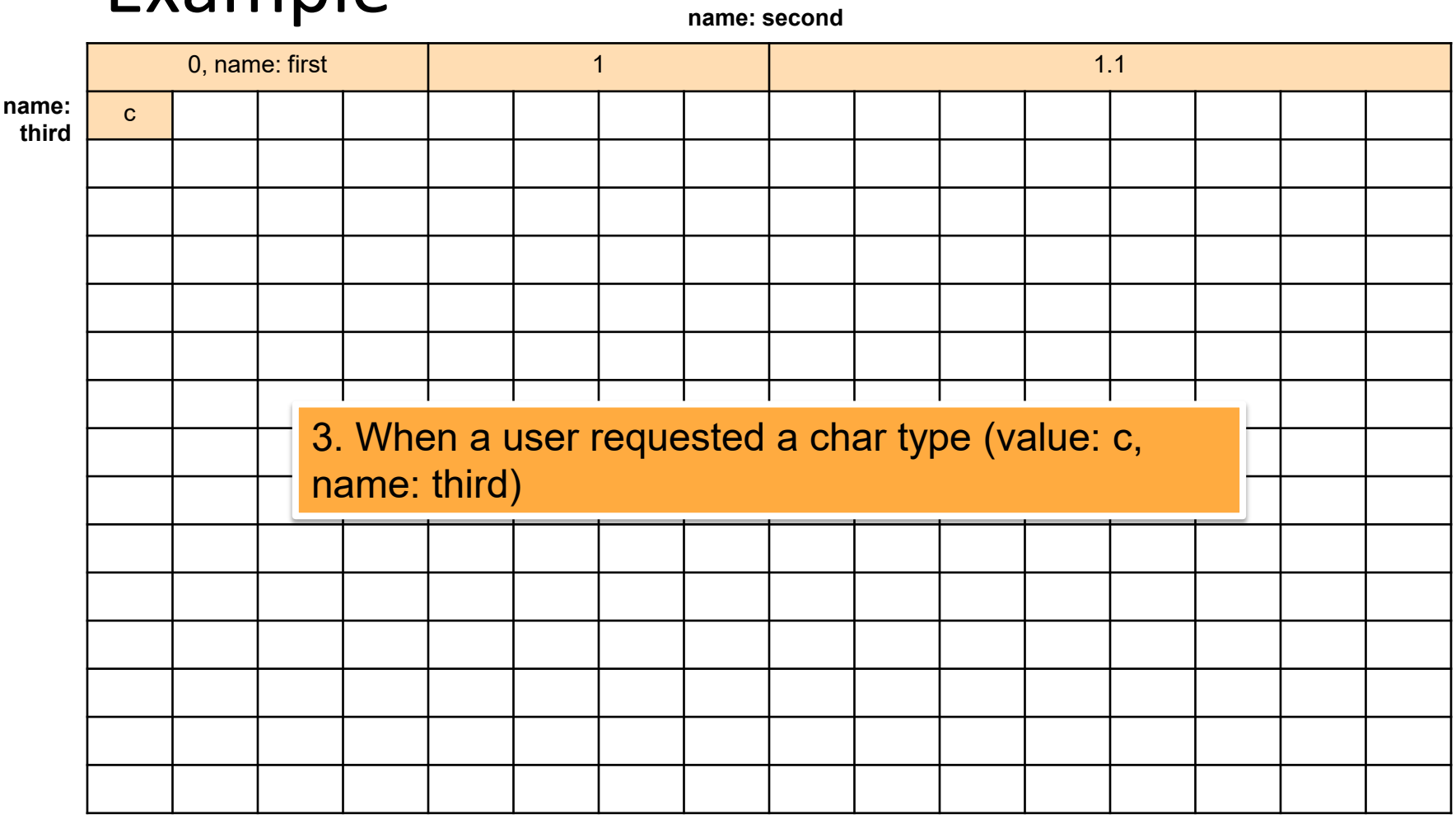
Example

name: second

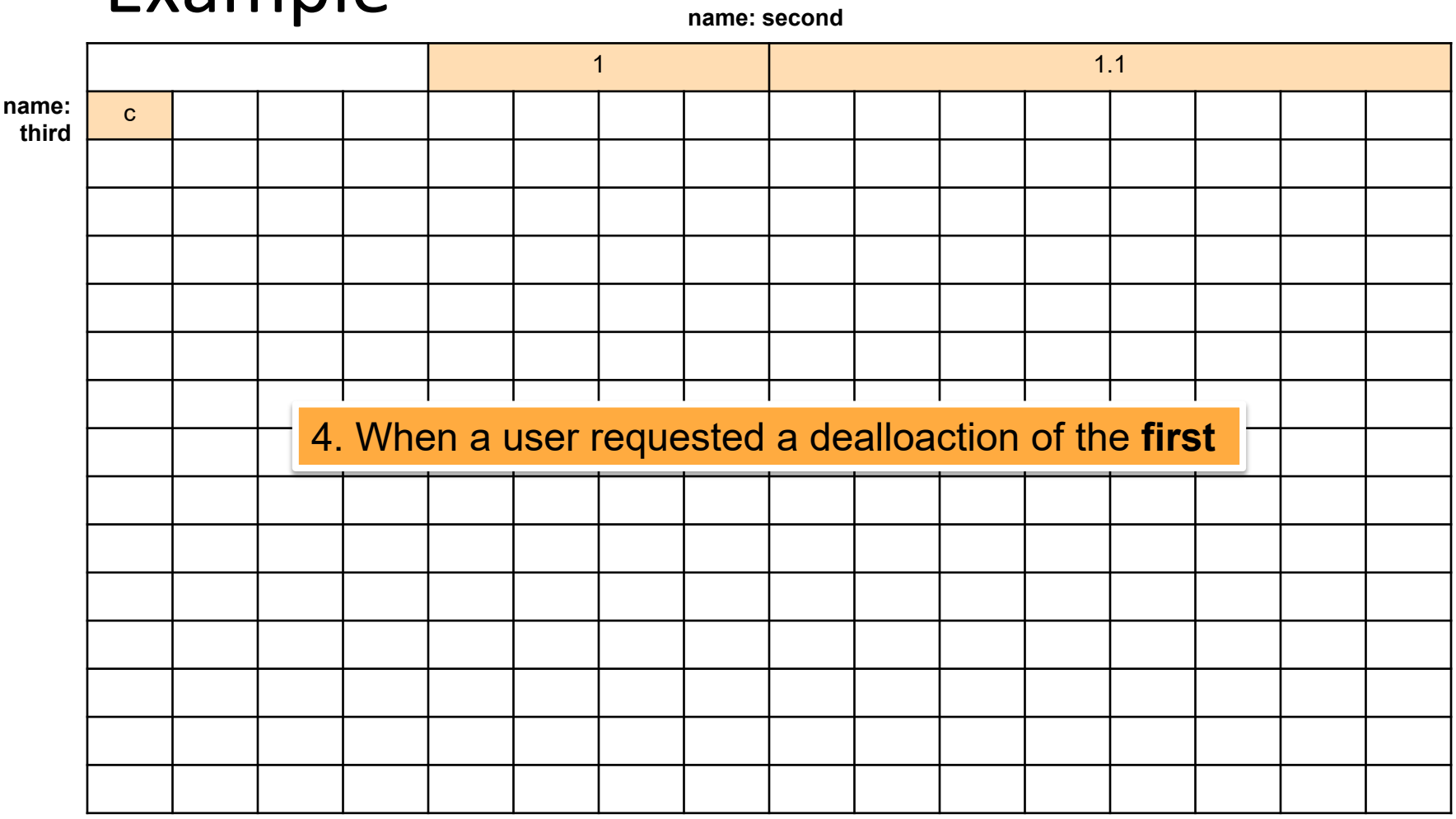
0, name: first	1	1.1

2. When a user requested a struct type for an int type and double type(value: 1, 1.1, name: second)

Example


name: second
0, name: first
1
1.1
name: third
c
3. When a user requested a char type (value: c, name: third)

Example


name: second
1
1.1
name: third
c
4. When a user requested a dealloaction of the **first**

Example

name: second

name: third

1	1.1	c			

5. After a user requested a dealloaction of the **first,** program move the other data to fill the hole

Types of data that your program supports

- Short($0 \sim 2^{15}-1$)
- Char
- float
- Long($0 \sim 2^{63}-1$)
- Int($0 \sim 2^{31}-1$)
- Struct(which has above 5 data types and specific values)

Additional condition

- Length of name: 50
- A struct can contain a maximum of 8 types
  (e.g., 2 for int, 2 for short, and 4 for long)
- Struct cannot contain struct type
- You don’t need to consider the negative number

Invalid input(or exception) handling

We will only consider these cases as exceptions.

- Overflow of **short**, **int**, and **long** type: when user requests the input which exceeds the range of the data type (refer to the range of 3 types on page 12)
- Overflow of **struct**: when user requests struct which exceeds the available memory size
- **Invalid data type**: when user requests invalid data type or unsupported data type
- **Duplicated name**: when user requests duplicated name already stored in memory
- **Deallocating non-existent data**: when user requests a deallocation of non-existent data

In these cases, program will return to the beginning without performing any operations on invalid requests, while maintaining the existing memory state (See page 22).

If there is not enough memory for the requested data

It should not be allowed!

Your program just prints out a string

"**There is not enough memory for the data which you requre, you can only use X byte(s)**"

e.g., **There is not enough memory for the data, you can only use 1 byte(s)**

Printing format and explain again

When user requested an int type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
int int_a
Please input a value for the data type
32
There is memory dump!
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
```

Printing format and explain again

When user requested a char type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
char char_a
Please input a value for the data type
k
There is memory dump!
20 00 00 00 6b 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
char_a
```

Printing format and explain again

When user requested a float type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
float float_a
Please input a value for the data type
7.7
There is memory dump!
20 00 00 00 6b 66 66 f6 40 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
char_a
float_a
```

Printing format and explain again

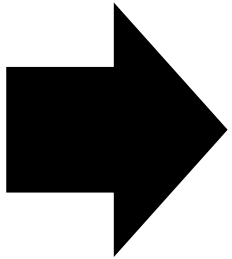
When user requested a struct type which has short and float data types

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
struct struct_a
How many data should be in the struct
2
Please input each type and its value
short 9
float 7.7
There is memory dump!
20 00 00 00 6b 66 66 f6 40 09 00 66 66 f6 40 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
char_a
float_a
struct_a
```

Printing format and explain again

When user requested a deallocation of float_a

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
struct struct_a
How many data should be in the struct
2
Please input each type and its value
short 9
float 7.7
There is memory dump!
20 00 00 00 6b 66 66 f6 40 09 00 66 66 f6 40 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
char_a
float_a
struct_a
```



```
Do you want to allocate data (1) or deallocate data (2) ?
2
Input the name of data you want to deallocate
float_a
float_a has been deallocated
There is memory dump!
20 00 00 00 6b 09 00 66 66 f6 40 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

-----Data you have now-----
int_a
char_a
struct_a
```

Printing format and explain again

When there is not enough memory for the data

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
short short_b
There is not enough memory for the data which you require, you can only use 0 byte(s)
```

Printing format and explain again

When there is invalid input or exception (e.g., overflow or invalid type)

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
int int_invalid
Please input a value for the data type
2147483648
There is invalid input
Do you want to allocate data (1) or deallocate data (2) ?
```

Invalid input (Exceeds the range of 'int')

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
shrt short_a
Invalid type
Do you want to allocate data (1) or deallocate data (2) ?
```

Invalid type

Use this dump_mem function!

```
void dump_mem(const void *mem, size_t len) {
    const char *buffer = mem;
    size_t i;
    for (i=0; i<len; i++){
        if (i>0 && i%16 == 0) {
            printf("\n");
        }
        printf("%02x ", buffer[i] & 0xff);
    }
    puts("");
}
```

Submission

- You should submit your code with a code description that explains your code (i.e., comments in the file). In the description, your code must be well commented to explain your algorithm. Make your code .zip file with ”StudentID_YourName.zip" and submit your .zip file on blackboard.

ex)

20195147_HongjunYang.zip

- assignment4.c // assignment4 code
- assignment4.h // assignment4 header file
- Makefile // Makefile
- report.pdf // assignment4 code description