Семинар #9: Память. Домашнее задание.

Задачи на работу с памятью

Задача 1 Как выглядит память, инициализируемая при создании следующих переменных (в системе с порядком байт Little Endian):

- `int a = 0x11223344;`
- `int b = 65535;`
- `int c = -1;`
- `int array[3] = {10, 2000, 65535};`
- `char str[8] = ''Hello'';`
- `float x = 1.0f;`
- ```
  struct data {
      char str[5];
      int number;
  };
  struct data c = {"Cat", 100000};
  ```

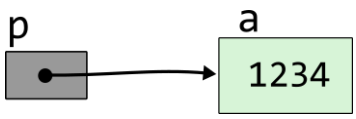
Память представить в виде последовательности 2-значных шестнадцатеричных чисел. Например число $123456 = 1e240_{16}$ будет храниться в памяти как `40 E2 01 00`.
Подсказка: Чтобы проверить, как будет выглядеть память, можно создать указатель типа `char*` на эту память и распечатать каждый байт в виде шестнадцатеричного числа:

```
char* p = (char*)&a;
for (int i = 0; i < sizeof(a); ++i) {
    printf("%02x ", p[i]);
}
```
```

Задачи на указатели

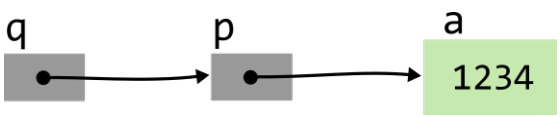
- **Указатель на `int`:** Удвойте значение переменной `a`, используя только указатель `p`.

```c
#include <stdio.h>
int main() {
    int a = 1234;
    int* p = &a;
    // Тут нужно написать 1 строку кода
    printf("%i\n", a);
}
```



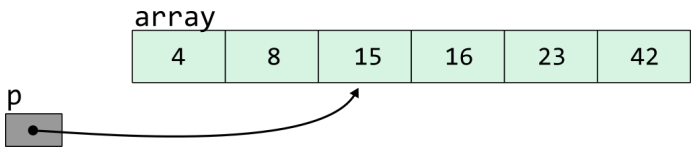
- **Указатель на указатель на `int`:** Удвойте значение переменной `a`, используя только указатель `q`.

```c
#include <stdio.h>
int main() {
    int a = 1234;
    int* p = &a;
    int** q = &p;
    // Тут нужно написать 1 строку кода
    printf("%i\n", a);
}
```



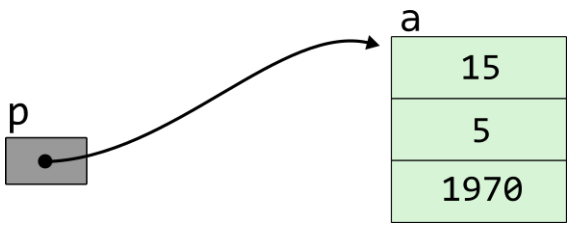
- **Указатель на элемент массива:** Удвойте значение `array[1]`, используя только указатель `p` (не меняя `p`, он должен указывать на `array[2]`).

```c
#include <stdio.h>
int main() {
    int array[6] = {4, 8, 15, 16, 23, 42};
    int* p = &array[2];
    // Тут нужно написать 1 строку кода
    printf("%i\n", a);
}
```



- **Указатель на структуру** Удвойте значение поля `year`, используя только указатель `p`.

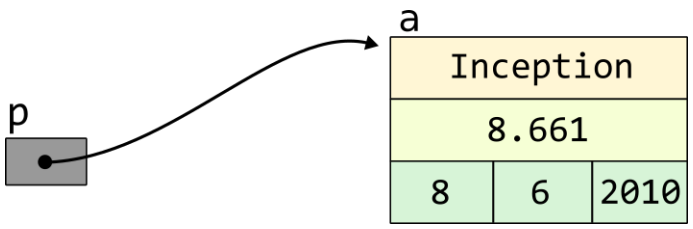
```c
#include <stdio.h>
struct date {
    int day;
    int month;
    int year;
};
int main() {
    struct date a = {15, 5, 1970};
    struct date* p = &a;
    // Тут нужно написать 1 строку кода
    printf("%d %d %d\n",
           a.day, a.month, a.year);
}
```



Указатель на структуру `Movie`

```c
#include <stdio.h>
struct movie {
    char title[50];
    float rating;
    struct date release_date;
};
typedef struct movie Movie;


int main() {
        Movie a = {"Inception", 8.661, {8, 6, 2010}};
        Movie* p = &a;
}
```

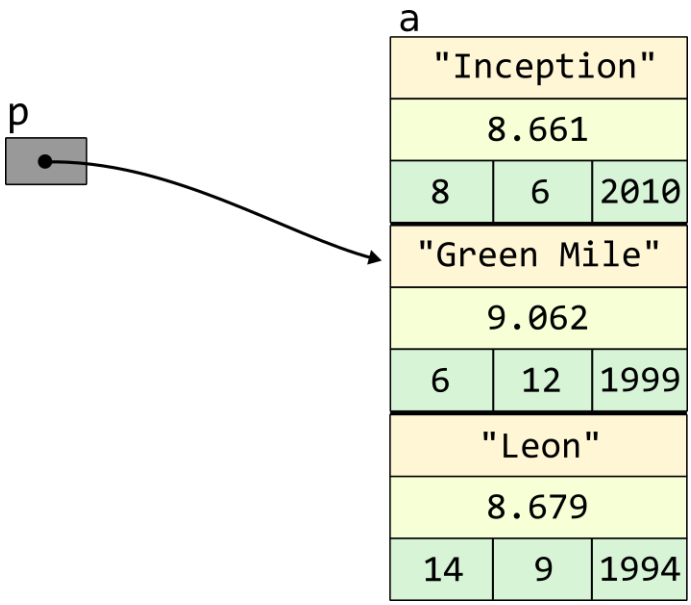


Задача #12: Удвойте значение поля `rating`, используя только указатель `p`.
Задача #13: Удвойте значение поля месяца выхода фильма, используя только указатель `p`.

Указатель на массив структур

```c
#include <stdio.h>
struct movie {
        char title[50];
        float rating;
        struct date release_date;
};
typedef struct movie Movie;

int main() {
        Movie a[3] = {{"Inception", 8.661, {8, 6, 2010}},
                      {"Green Mile", 9.062, {6, 12, 1999}},
                      {"Leon", 8.679, {14, 9, 1994}}};
        Movie* p = &a[1];
}
```



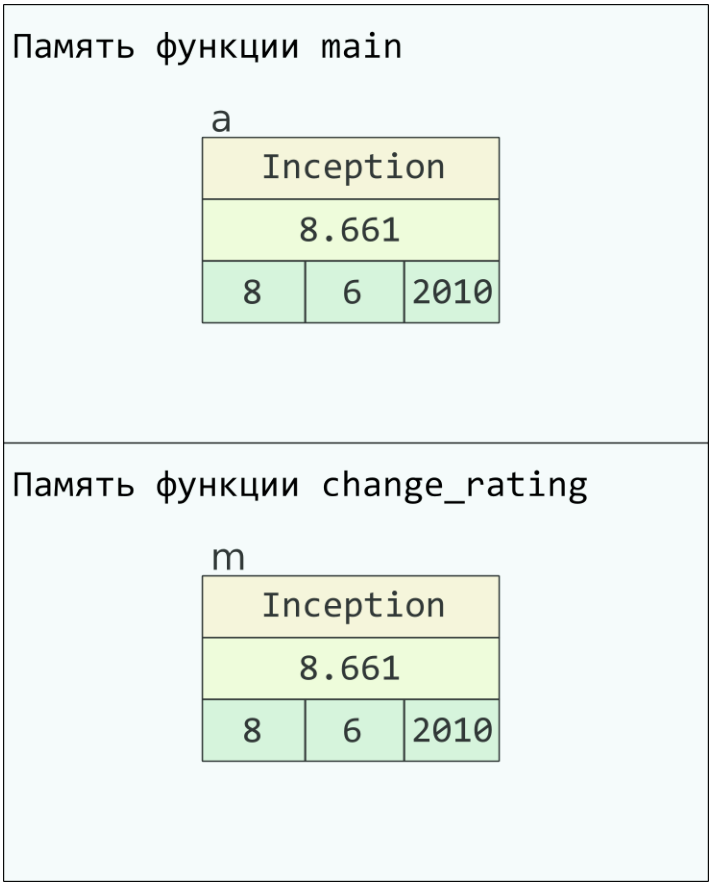
Задача #14: Удвойте значение рейтинга фильма `Inception`, используя только указатель `p`. (не меняя `p`, он должен указывать на `a[1]`)
Задача #15: Удвойте значение года выхода фильма `Leon`, используя только указатель `p`. (не меняя `p`, он должен указывать на `a[1]`).

Передача в функцию по значению

```c
#include <stdio.h>

struct movie {
        char title[50];
        float rating;
        struct date release_date;
};
typedef struct movie Movie;

void change_rating(Movie m) {
        m.rating += 1;
}
int main() {
        Movie a = {"Inception", 8.661,
                        {8, 6, 2010}};
        change_rating(a);
}
```



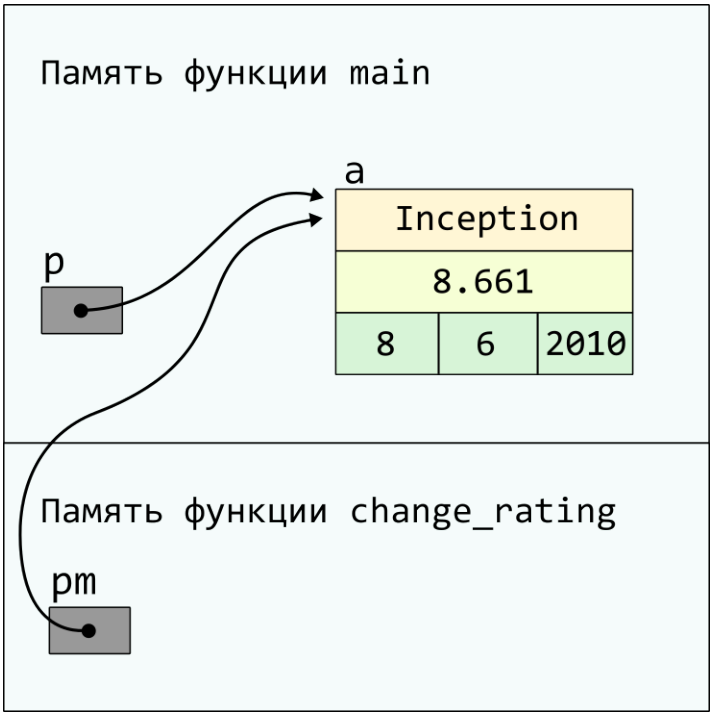
Всё, что передаётся в функцию, копируется. Поэтому функция `change_rating` будет менять поле `rating` у копии структуры `a`, а изначальная структура не изменится.

Передача в функцию по указателю:

```c
#include <stdio.h>

struct movie {
        char title[50];
        float rating;
        struct date release_date;
};
typedef struct movie Movie;

void change_rating(Movie* pm) {
        pm->rating += 1;
}
int main() {
        Movie a = {"Inception", 8.661,
                        {8, 6, 2010}};
        Movie* p = &a;
        change_rating(&a);
}
```



Всё, что передаётся в функцию, копируется. Но теперь туда копируется указатель, который содержит адрес стурктуры `a`. Используя этот указатель, мы можем изменить изначальную структуру. Более того, так как указатель занимает меньше память, его копирования в функцию происходит быстрее, чем копирование всей структуры.

Задача #16: Напишите функцию `change_day(struct date* pd)`, которая будет увеличивать день на `1`. Используйте эту функцию, чтобы увеличить день даты выхода на `1` у структуры `a`.