

### Задача 1:

1. Да се напише структура "Book", която съдържа следната информация(полета):

- име на книгата
- име на автора
- жанр
- цена
- брой продажби
- други по ваш избор

// Определете кои са подходящите типове за полетата на тази структура.

2. Напишете функция, която прочита въведени стойности и ги задава като стойности на полетата на подаден обект от горния тип (използвайте динамична памет). Напишете функция, която освобождава динамичната памет преди приключване на програмата.

```
void initialize(Book& b);  
void deleteDynamic(Book& b);
```

3. Въведете масив от n на брой книги, като n се въвежда от клавиатурата, използвайте горната функция.

4. Напишете функция, която отпечатва информацията от даден обект от горния тип.

```
void print(const Book& b);
```

5. Напишете функции, които отпечатват книгата с най-ниска и с най-висока цена:

- void printBookWithLowestPrice(Book\* books, size\_t count);
- void printBookWithHighestPrice(Book\* books, size\_t count);

6. Напишете функция, която намира книгата с най-много продажби:

```
Book& getBestBook(Book* books, size_t count);
```

### Задача 2:

1. Да се напише структура "Animal", която съдържа следните полета:

- вид на животното
- име на животното
- години на животното
- други по ваш избор

2. Напишете функцията, която въвежда n на брой инстанции на Animal (да се използва динамична памет).

3. Напишете структура "Zoo", която съдържа:

- име на зоологическата градина
- списък от животни
- брой на животните
- максимален брой на животните (максимален капацитет)

4. Напишете функция, която инициализира зоологическата градина:

```
void initialize(Zoo& z, char* name, Animal* a, size_t count, size_t MAX_COUNT);
```

5. Напишете функция, която проверява дали има животни от подаден вид(като параметър на функцията) в зоологическата градина:

```
bool hasSpecies(const Zoo& z, char* species);
```