Указатели

В следващите задачи, под това да подаваме масив на функция, ще разбираме да подаваме указател към първия му елемент и указател след последния му елемент.

Задача 00 - Принтиране

```
void print(const int* begin, const int* end);
```

Напишете функция, която принтира масив.

Задача 01 - Upper Bound

```
const int* upperBound(const int* begin, const int* end, int element);
```

Напишете функция, която по подаден сортиран масив и число, връща указател към първия елемент от масива, който е по-голям от подаденото число. Ако такъв елемент не съществува, функцията да връща *nullptr*.
Пример:

```
const int arr[] = {0, 1, 2, 5, 7};
int *begin = &arr[0];
int *end = &arr[4]
const int *element = upperBound(begin, end, 4);
const int *notFound = upperBound(begin, end + 2, 4);

print(element, end); // -> 5 7
if(!notFound) {
  cout << "Not␣Found\n"; // -> Not Found
}
```

Задача 02 - Еднакви масиви

```
bool equal(const int* firstBegin, const int* firstEnd, const int*
   secondBegin, const int* secondEnd);
```

Напишете функция, която проверява дали елементите на два масиви съвпадат.
Пример:

```
const int arr[] = {1, 2, 1, 3};
const int arr2[] = {1, 2};
int *begin = &arr[0];
int *end = &arr[3];
```

```
int *begin2 = &arr2[0];
int *end2 = &arr2[1];

cout << equal(begin, begin + 2, begin2, end2) << '\n'; // -> true
cout << equal(begin, end, begin2, end2); // -> false
```

Задача 03 - Замяна

```
void replace(int* begin, int* end, int oldValue, int newValue);
```

Напишете функция, която заменя всички срещания на елемент в масив с подадена нова стойност.
Пример:

```
int arr[] = {0, 9, 2, 9, 3, 9};
int *begin = &arr[0];
int *end = &arr[5];
replace(begin, end, 9, 1);
print(begin, end); // -> 0 1 2 1 3 1
```

Задача 04 - Завъртане

```
void rotate(int* begin, int* mid, int* end);
```

Напишете функция, която по подаден масив и указател *mid* към елемент в него, разменя елементите в интервала [mid, end) с тези в интервала [begin, mid).
Пример:

```
int arr[] = {0, 1, 2, 3, 4};
int *begin = &arr[0];
int *end = &arr[4];
rotate(begin, begin + 2, end);
print(begin, end); // -> 2 3 4 0 1
```

Задача 05 - Търсене

```
const int* search(const int* firstBegin, const int* firstEnd, const int
   * secondBegin, const int* secondEnd);
```

Напишете функция, която по подадени 2 масива, връща указател към първия елемент в първия масив, от който започва да се среща втория. Ако няма такъв елемент, да се върне *nullptr*.
Пример:

```
const int arr[] = {0, 1, 2, 1, 2};
const int seq[] = {1, 2};
const int* begin = search(cbegin(arr), cend(arr), cbegin(seq), cend(seq
   ));
print(begin, cend(arr)); // -> 1 2 1 2
```

cbegin(), cend() връщат указатели към началото и края на масива.

Задача 06 - Постфикс

```
bool endsWith(const int* firstBegin, const int* firstEnd, const int*
    secondBegin, const int* secondEnd);
```

Напишете функция, която по подадени 2 масива, проверява дали последните елементи на първия съвпадат с тези на втория.
Пример:

```
const int arr[] = {2, 4, 5, 3, 7, 6};
const int arr2[] = {7, 6};

cout<<endsWith(cbegin(arr), cend(arr), cbegin(arr2), cend(arr2)) << '\n
    '; // -> true
cout<<ends_with(cbegin(arr), cbegin(arr) + 5, cbegin(arr2), cend(arr2))
    ; // -> false
```

Задача 07 - Последно срещане

```
const int* find_end(const int* first_begin, const int* first_end, const
     int* second_begin, const int* second_end);
```

Напишете функция, която по подадени 2 масива, връща указател към последния елемент в първия, от който започва да се среща втория. Ако няма такъв елемент, да се върне *nullptr*.
Пример:

```
const int arr[] = {0, 1, 2, 1, 2, 5};
const int seq[] ={1, 2};
const int* begin = find_end(cbegin(arr), cend(arr), cbegin(seq), cend(
    seq));
print(begin, cend(arr)); // -> 1 2 5
```

Задача 08 - Уникалност

```
int* unique(int* begin, int* end);
```

Напишете функция, която заменя всяка поредица от повтарящи се елементи в масив само с един от този елемент. Функцията да връща новия логически край на масива.
Пример:

```
int arr[] = {1, 2, 1, 1, 3, 3, 3, 4, 4, 5, 4};
int* end = unique(begin(arr), end(arr));
print(cbegin(arr), end); // -> 1 2 1 3 4 5 4
```

Задача 09 - Разделяне

```
const int* partition(int* begin, int* end, int element);
```

Напишете функция, която по подаден масив и естествено число пренарежда масива, така че в началото му да са всички елементи, по-малки от подаденото число, а в края - всички по-големи или равни. Функцията да връща указател към началото на втората група. Редът на елементите в двете групи няма значение.

Пример:

```
int arr[] = {4, 7, 1, 3, 2, 6, 5, 9, 0};
const int *mid = partition(begin(arr), end(arr), 5);

print(begin(arr), mid); // -> 4 0 1 3 2
print(mid, cend(arr)); // -> 5 9 6 7
```

Задача 10* - Следваща пермутация

```
bool next_permutation(int* begin, int* end);
```

Напишете функция, която нарежда елементите в масив в тяхната следваща пермутация (спрямо лексикалната им наредба). Ако е достигната последната пермутация (т.е. масивът е сортиран в низходящ ред), то той да се пренареди до първата му пермутация (т.е. да е сортиран във възходящ ред) и функцията да върне *false*. Иначе функцията трябва да връща *true*.

Пример:

```
int arr[] = {1, 2, 3};
do
{
    print(cbegin(arr), cend(arr));
    cout << endl;
} while(next_permutation(begin(arr), end(arr)));
/*
    1 2 3
    1 3 2
    2 1 3
    2 3 1
    3 1 2
    3 2 1
*/
```