УПРАЖНЕНИЕ 10

Въпроси: 1. Какви типове памет има и кое къде живее?

2. Какво е указател? Къде живее той? Какво казахме за правилно използване ?

3. int *p, int *q; Какво ще върне? if(*p == *q) if(p==q) if(!p)

4.Какво знаем за константните променливи? Можем ли да имаме константи указатели?

- **Типове указатели**

-Константен указател

Синтаксис:

<тип>* const <име на променлива> = <стойност>

-Указател към константи

Синтаксис:

const <тип>* <име на променлива> = <стойност>

-Константен указател към константи

const <тип>* const <име на променлива> = <стойност>

-Референции / int& ref ⇔ int* const ref/ syntax sugar

Референциите главно се използват за предаване на параметри на функции и връщане на стойности от функции. Референцията не е копие на променливата, към която се отнася. **Тя всъщност е същата променлива, но под друго име!**

Пример: Функцията swap;

- **Масиви и указатели**

- адресна аритметика:
  - arr[10] = 10; ⇔ *(arr+10) = 10; 0x..0, 0x..04
    int matrix[n][m] n,m > 10
  - matrix[5][7] ⇔ *(matrix +5*m + 7);

Какво прави arr++;

Пример: print(int * arr, int size);

- **Динамична памет**
- Заделяне

Синтаксис:
<тип> * <име_на_променлива> = new <тип>;
int* p = new int;

<тип> * <име_на_променлива> = new <тип> [<брой елемент>];
int* pArr = new int[size];

При заделяне, добавяме модификатора (std::nothrow) , за да може да обработим случаите, в които заделянето е неуспешно! т.е.

<тип> * <име_на_променлива> = new(std::nothrow) <тип> [<брой елемент>]

Накрая проверяваме дали указателят, който сочи към паметта е празен(NULL).

Примери: заделяне на памет по въведено n от потребителя.

- Освобождаване / длъжност на програмиста в C/C++ е да се погрижи заделената от него динамична памет да бъде освободена /
Синтаксис:

delete[] <име_на_променлива>;
delete <име_на_променлива>;

Какво представлява Memory leak, как може да се получи?

пример: при заделяне на паметта, директно да пренасочим указателя (губим старата памет, няма откъде да откопираме елементите)
пример: да вървиш по оригиналния указател и да загубиш указателя към началото;
Да реализираме pushBack и insertAt на динамично заделен масив в main-а.
Какво ще стане, ако изнесем имплементацията във функция?

- използване на референции към указатели като параметри на функции
  практическа полза - когато искаме да пренасочваме указател, дошъл като параметър в рамките на една функция и искаме да променим същинския указател, а не неговото локално копие използваме <тип>*&

За пркатически примери и допълнителни обяснения разгледайте примерите в репото : https://github.com/VasiPeycheva/UP--2017-2018