# САОД. Задание 2

## Темы

* указатели и ссылки,
* массивы,
* z-строки,
* передача параметров.

## Задания

1. Запустить Visual Studio и создать на рабочем столе проект “pointer” (решение Ex2).
   1. После **#include** <iostream> добавить строку **using namespace std;**
   2. В main ввести код:  
      int x = 3;  
      int \*p = &x;   
      cout << x << ' ' << \*p << ' ' << p << endl;
   3. Убедиться, что х и \*p равны 3 и 3 соответственно, а p – адрес x, т.е. \*p получили значение.
   4. Добавить строчки p++; cout << p << endl;
   5. Убедиться, что p++ передвинула адрес на 4 байта (размер целого).
   6. Добавить cout << p - &x<< endl;   
      Убедиться, что результат (разность двух указателей) равен 1 (между ними 1 целое число);
2. Следующее упражнение трудно сделать без указателей.   
   Посмотрим, как целое число размещено в памяти по байтам. Преобразуем тип указателя к byte\* и выведем байты в цикле:  
   typedef unsigned char byte; // Нет встроенного byte

byte \*pb = (byte \*)--p; // вернули указатель на x и преобразовали

for (byte \*pt = pb; pt - pb<sizeof(int); pt++)

cout << (int)\*pt << ' '; // выводим как int

cout << endl;

Разобрать текст. Задать вопросы, если что-то непонятно или рассказать преподавателю как понимаете текст.

Обратите внимание, что целое 3 расположено в памяти не как 0 0 0 3, а как 3 0 0 0. Расположение в памяти зависит от платформы.

1. **Самостоятельно**. Разработать функцию Swap с использованием указателей. Ваше понимание проверяйте, рассказывая его преподавателю.
   1. Закомментируйте весь код в main.
   2. Объявите 2 целочисленных переменных int a = 3, b=5; и прямо в main напишите код, который приведет к обмену переменных значениями.
   3. Убедитесь, что он работает - до обмена a = 3, b = 5, после обмена a = 5, b = 3.
   4. Добавьте функцию типа   
      void Swap(int a, int b)   
      перед main и перенесите в нее код обмена значениями.
   5. Вызовите ее в main и убедитесь, что она не работает. Подумайте почему.

Очевидно, что если бы в функцию попали адреса фактических параметров, то задачу бы выполнить удалось.

* 1. Самостоятельно доработайте функцию, чтобы оп-на принимала параметры-указатели void Swap(int \*pa, int \*pb).
  2. Убедитесь, что теперь функция работает. Вызывать конечно нужно, передавая в нее адреса переменных (&a, &b).
  3. Вспомните (по лекции) про параметры, которые передаются по ссылке.   
     Измените заголовок функции на void Swap(int &a, int &b) и внесите необходимые изменения в ее реализацию (теперь при вызове указываем только имена переменных, а не адреса).   
     Убедитесь, что так удобнее и более естественно, а функция работает.

Теперь понятно, как работают указатели.

Если времени мало, то п. 4 можно пропустить.

1. Переносим функцию Swap в отдельный модуль (utils.cpp и utils.h)
   1. Добавляем оба файла в проект;
   2. В utils.h переносим прототип функции, а в utils.cpp реализацию;
   3. В utils.cpp первой строкой добавляем #include “stdafx.h” (если проект сконфигурирован на использование прекомпилированных заголовков).
   4. В основном модуле включаем заголовок “utils.h”;
   5. Проверяем, что все работает, или добиваемся, чтобы работало.
2. Знакомство с массивами.
   1. Добавить в Решение новый консольный проект arrays;
   2. В main объявить массив и вывести его на консоль в одну строку  
      int a[5]{1,2,3,4,5};  
      for(int i = 0; i<5; i++)  
       cout<<a[i] << ' ';  
      cout<<endl;
   3. Вывести массив с помощью указателя на целое:

for (int \*p = a; p-a<5; p++)

cout << \*p << ' ';

cout << endl;

* 1. Разберитесь как работает цикл! Что обеспечивает инициализатор? В чем заключается условие и почему оно правильное?

1. Передача массива как параметр:
   1. Создать функцию, которая выполняет циклическую перестановку массива   
      void rotate (int a[], int n) { int x = a[0];

for (int i = 1; i < n; i++)

a[i - 1] = a[i];

a[n - 1] = x;}

* 1. Проверить, что функция работает или добиться, чтобы работала!

Обратите внимание, что при передаче массива в функцию нужно указывать его размер. Имя массива – это просто указатель на блок памяти и размер его не известен!

* 1. Если есть время, то реализуйте функцию через указатели без использования целочисленного счетчика i.

1. Наша функция переставляет массив против часовой стрелки.

Самостоятельно доработайте функцию, так чтобы направлением перестановки можно было управлять. void rotate (int a[], int n, bool clockwise=false). Если последний параметр true, то вращать массив по часовой стрелке.

Убедитесь, что функция работает в обоих направлениях.

1. Z-строки.
   1. Добавить в Решение новый консольный проект strings.
   2. Объявить строку вывести ее на консоль (встроенного типа string в C++ нет. Используется массив символов)   
       char str[] = "Hello!";

cout << str << endl;

* 1. Вывести на консоль размер массива и длину строки:  
     cout << strlen(str) << ' ' << sizeof(str) << endl;

Обратите внимание, что они отличаются на 1. Почему это произошло?

* 1. Создайте дополнительный массив символов и скопируйте в него строку:   
     char t[32];   
     for (char \*pd = t, \*ps = str; \*pd++ = \*ps++;);
  2. Убедитесь, что произошло копирование. Разберитесь по тексту как происходит копирование. Перепишите цикл менее компактно, но более понятно.
  3. Вынесите текст в функцию void string\_copy(char \*d, char\*s);

1. Если есть время, вынесите функцию в utils.
2. Реализуйте самостоятельно функции:
   1. int len(const char \*s); - должна возвращать длину строки
   2. int compare(const char \*s, const char \*t);   
      Функция должна возвращать целое > 0, если s > t, < 0, если s < t и 0, если s == t;

Если не успеваете, то проделайте эти упражнения дома.

Завершайте работу, удалите ненужные файлы, архивируйте решение - это ваш второй трофей в C++.

Теперь мы умеем:

* Объявлять и использовать указатели;
* Передавать параметры по ссылке;
* Объявлять и использовать массивы;
* Работать с z-строками (*сами этого делать, как правило, не будем, есть много стандартных функций, но, в принципе, умеем*).

# Отчет

Подготовьте в Word краткий (*не более 1 стр.*) отчет о выполнении задания.

В отчете опишите: какие действия над указателями и адресами вы проделали в Задании, как передается массив в функцию, каков принцип работы с z-строками.

Отчет и архив решения поместите как ответ на задание.