

ПРАКТИЧЕСКАЯ РАБОТА №3. УКАЗАТЕЛИ

Задание 1.

Создайте новый проект, скопируйте в файл *main.c* содержимое файла *pr_5.c*. Запустите программу, проанализируйте результаты:

- посмотрите, по каким адресам располагаются переменные, какой объем памяти они занимают и начертите схематичное расположение элементов в памяти;
- отметьте стрелками, куда указывают указатели;
- отметьте, куда стали указывать указатели после присваивания им значения указателя *p2*, обведите (подчеркните, раскрасьте или как-то еще) адресуемые области для каждого указателя, объясните, как получаются выведенные программой значения;
- отметьте, куда стали указывать указатели *p1* и *p3* после применения к ним операций инкремента и декремента, обведите (подчеркните, раскрасьте или как-то еще) адресуемые области указателей *p1* и *p3*, объясните, как получаются выведенные программой значения;
- отметьте, куда стали указывать указатели *p1* и *p3* после последнего изменения, обведите (подчеркните, раскрасьте или как-то еще) адресуемые области указателей *p1* и *p3*, объясните, как получаются выведенные программой значения;
- сформулируйте общие выводы, к которым вы пришли в процессе выполнения задания.

Задание 2.

Создайте новый проект, скопируйте в файл *main.c* содержимое файла *pr_6.c*. Проанализируйте текст программы, найдите в нем синтаксические ошибки и исправьте их, добейтесь, чтобы компилятор не выводил никаких предупреждений. В начало программы добавьте вывод на экран адресов всех переменных (помним: указатель – тоже переменная 😊), а в конец – значений всех переменных. Запустите программу, проанализируйте результаты ее работы так же, как при выполнении первого задания, объясните, почему они именно такие. Замените инструкцию-выражение «*m += 2;*» инструкцией-выражением «*m++;*», запустите программу еще раз, проанализируйте результат.

Задание 3.

Создайте новый проект. Объявите по две переменные типов *char*, *int* и *double*, а также указатель на *char*. Выведите на экран размеры и адреса всех переменных, начертите схему расположения переменных в памяти. Поменяйте порядок объявления переменных (например, *int*, *char*, *double*, *char**, *char*, *double*, *int*). Запустите программу повторно, проанализируйте, что изменилось. Задайте переменной типа *int* такое значение, чтобы значение каждого байта было уникальным, используйте для этого шестнадцатеричную константу. Запишите адрес этой переменной в указатель на *char* и с его помощью выведите на экран содержимое каждого байта

(тоже в шестнадцатеричной системе счисления). Проанализируйте, прямой или обратный порядок расположения байт при записи числа применяется в используемой вами системе.

Полностью повторите выполнение этого задания на компьютере с другой операционной системой и/или другой IDE (можно использовать домашний компьютер или онлайн-компилятор). Сравните результаты работы программы на разных платформах, сделайте выводы.