

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, например

адреса	EEBB3884	EEBB3885	EEBB3886	EEBB3887	EEBB3888	EEBB3889	EEBB388A	EEBB388B	EEBB388C	EEBB388D	EEBB388E	EEBB388F	EEBB3890	EEBB3891	EEBB3892	EEBB3893	EEBB3894	EEBB3895	EEBB3896	EEBB3897	EEBB3898	EEBB3899	EEBB389A	EEBB389B	EEBB389C	EEBB389D	EEBB389E	EEBB389F	EEBB3900	EEBB3901	EEBB3902	EEBB3903
переменные	a				b				c				p1				p2				p3				p4							
значения переменных	1				2				3				EEBB3884				EEBB3888				EEBB3884				NULL							

- отметьте, куда указывают указатели (запишите значения указателей, закрасьте одним цветом указатель и его адресуемую область), например

адреса	EEBB3884	EEBB3885	EEBB3886	EEBB3887	EEBB3888	EEBB3889	EEBB388A	EEBB388B	EEBB388C	EEBB388D	EEBB388E	EEBB388F	EEBB3890	EEBB3891	EEBB3892	EEBB3893	EEBB3894	EEBB3895	EEBB3896	EEBB3897	EEBB3898	EEBB3899	EEBB389A	EEBB389B	EEBB389C	EEBB389D	EEBB389E	EEBB389F	EEBB3900	EEBB3901	EEBB3902	EEBB3903
переменные	a			b			c						p1		p2			p3			p4											
значения переменных	1			2			3						EEBB3884		EEBB3888			EEBB3884			NULL											
адресуемые области указателей и значения в адресуемых областях	1			2																												
	4.2439915824246e-314																															

- отметьте, куда стали указывать указатели после присваивания им значения указателя p2, обозначьте адресуемые области для каждого указателя (закрасьте одним цветом указатель и его адресуемую область), объясните, как получаются выведенные программой значения;

- отметьте, куда стали указывать указатели p1 и p3 после применения к ним операций инкремента и декремента, обозначьте адресуемые области указателей p1 и p3, объясните, как получаются выведенные программой значения;

- отметьте, куда стали указывать указатели p1 и p3 после последнего изменения, обозначьте 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.

Важно! Задание выполняется или на компьютерах под управлением ОС семейства Windows, или с использованием онлайн-IDE Replit <https://replit.com/languages/c>

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 😊) и объемов занимаемой ими памяти, а в конец – значений всех переменных. Поставьте точку останова программы на строке ввода данных. Запустите программу в режиме отладки. Откройте окно «Watches» для наблюдения за значениями переменных в процессе работы программы. Проанализируйте, что происходит в процессе работы программы, объясните, почему результаты 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 другой платформе (можно использовать автономные IDE под управлением различных операционных систем или онлайн-IDE), возможные пары:

- первое выполнение программы осуществляется под управлением ОС Kubuntu, второе – под управлением ОС Windows,
- первое выполнение программы осуществляется под управлением ОС Kubuntu, второе – в онлайн-IDE Replit <https://replit.com/languages/c>,
- первое выполнение программы осуществляется под управлением ОС Windows, второе – в онлайн-IDE GDB https://www.onlinegdb.com/online_c_compiler,
- первое выполнение программы осуществляется в онлайн-IDE Replit <https://replit.com/languages/c>, второе – в онлайн-IDE GDB https://www.onlinegdb.com/online_c_compiler.

Сравните результаты работы программы на разных платформах, сделайте выводы.