Темы

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

Действия над указателями и адресами:

Получение адреса переменной: использовался оператор & для получения адреса переменной (int \*p = &x).

Разыменование указателя: использовался оператор \* для получения значения, хранящегося по адресу, на который указывает указатель (cout <<\*p).

Арифметика указателей: выполнялись операции инкремента (p++) и вычитания (p - &x) для перемещения по памяти и определения расстояния между указателями. Важно, что эти операции выполняются в единицах типа данных, на которые указывает указатель.

Преобразование типа указателя: указатель типа int\* преобразовывался к типу byte\* ((byte \*)--p ) для доступа к отдельным байтам переменной int в памяти.

Передача массива в функцию:

Массивы в C++ всегда передаются в функцию как указатели на первый элемент массива. Это означает, что изменения, внесенные в массив внутри функции, отражаются на исходном массиве. Для корректной работы с массивом в функции необходимо также передавать его размер, так как функция получает только адрес начала массива и не имеет информации о его границах. Пример: void rotate(int a[], int n) – a является указателем на int, а n содержит размер массива.

Принцип работы с z-строками (C-style строками):

C-style строки (z-строки) представляют собой массивы символов, завершающиеся нулевым символом (\0). Функции для работы со строками (например, strlen, strcpy) определяют конец строки по этому нулевому символу. Важно учитывать, что sizeof возвращает размер массива, включая нулевой символ, а strlen возвращает только количество символов до нулевого символа (не включая его). При копировании строк необходимо следить за тем, чтобы буфер назначения был достаточно большим для хранения копируемой строки и завершающего нулевого символа. Функции string\_copy, len и compare демонстрируют основные операции над C-style строками.