

ROT: Циклический сдвиг

Задача: написать программу, циклически сдвигающую элементы массива на заданное количество позиций влево несколькими способами. Напишем три версии такой сдвигающей функции с одинаковым прототипом:

```
void Shift(int arr[], int size, int delta);
```

Пример: при сдвиге {7, 14, -2, 1, 9} на две позиции массив примет следующий вид: {-2, 1, 9, 7, 14}.

Пишем по шагам:

0. **Подготовьтесь к работе (0 баллов).** Заведите массив длины N=9. Заполните его числами по возрастанию от 1 до N (автоматически, без участия пользователя!). Выведите массив на экран. Разумеется, оформите заполнение и печать массива в виде функций

```
void Fill(int arr[], int size);  
void Print(int arr[], int size);
```

1. **Сдвиг по одному шагу (+1 балл).** Напишите функцию, сдвигающую элементы массива на одну позицию влево.

```
void Shift1(int arr[], int size);
```

Используйте вспомогательную переменную и копирование элементов налево в цикле. С её помощью напишите функцию, сдвигающую на несколько позиций.

```
void Shift_Ver1(int arr[], int size, int delta);
```

Прочитайте с клавиатуры число величину сдвига — количество позиций, на которое надо сдвинуть массив. Выполните сдвиг исходного массива, напечатайте результат. *Обратите внимание* и обработайте случаи $\Delta = 0$ и $\Delta \geq N$.

Введите сдвиг: 3

Способ 1:

```
1 2 3 4 5 6 7 8 9  
4 5 6 7 8 9 1 2 3
```

2. **Сдвиг разворотами (+1 балл).** Напишите функцию, разворачивающую задом наперёд часть массива, заданную индексами левой и правой границы. Элементы массива вне этих границ не должны портиться.

```
void Reverse(int arr[], int left, int right);
```

С её помощью напишите функцию Shift_Ver2, сдвигающую массив на несколько позиций, развернув сначала весь массив целиком, а затем левую и правую его части по отдельности. Напечатайте результат для этого способа сдвига.

Способ 2:

```
1 2 3 4 5 6 7 8 9
4 5 6 7 8 9 1 2 3
```

Пояснение:

```
1 2 3 | 4 5 6 7 8 9
9 8 7 6 5 4 | 3 2 1
4 5 6 7 8 9 | 1 2 3
```

3. (*) **Сдвиг в кольцах (+1 бонус)**. Написать функцию, которая сдвигает подмножество элементов массива, цепочку с шагом в Δ , начиная с элемента i_0 , ставя каждый элемент на своё место за один шаг.

```
void ShiftLoop(int arr[], int size, int delta, int i0);
```

Реализовать следующий алгоритм (который в некоторых случаях сдвинет весь массив, напр. при $\Delta = 2, 4, 5, 7$ для $N = 9$):

- запомнить выбранный элемент i_0 во временной переменной;
- поставить на его место элемент, находящийся правее на Δ позиций (по модулю, с учётом заикленности массива);
- перейти к образовавшейся “дырке” и заполнить её аналогично, повторять до тех пор, пока не вернёмся к исходной позиции i_0 ;
- заполнить последнюю дырку из временной переменной.

Напечатайте результат для этого способа сдвига, хоть он и неполноценный.

Способ 3а (только одно кольцо!):

```
1 2 3 4 5 6 7 8 9
4 2 3 7 5 6 1 8 9
```

Эта версия сдвигает массив целиком только в том случае, если N и Δ взаимно просты. В противном случае, цикл заканчивается после $\text{НОК}(N, \text{delta})$ шагов, а циклов таких $\text{НОД}(N, \text{delta})$.

Напишите функцию `void Gcd(int a, int b)`, вычисляющую НОД двух чисел. С её помощью напишите функцию `Shift_Ver3`, повторив $\text{НОД}(N, \text{delta})$ раз вызов функции `ShiftLoop`.

Напечатайте результат.

Способ 3:

```
1 2 3 4 5 6 7 8 9
4 5 6 7 8 9 1 2 3
```