

ADD: Сложение в столбик

Задача: написать программу, складывающую в столбик два числа, с контролем правильности результата (мы же знаем верный ответ). Примерная схема вычислений:

```
--> a -> A[] --o
      +          |---> C[] --> c' -->
--> b -> B[] --o
      |
      o-----> c" -->
```

То есть, прочитать с клавиатуры числа a, b, разобрать их на цифры в массивы A[], B[], сложить массивы, полученный результат C[] собрать обратно в число c' (первый способ). Для проверки сложить исходные числа и сравнить результат c" с полученным алгоритмом в столбик.

Пишем по шагам:

1. **Разберите на цифры (+1 балл).** Прочитайте с клавиатуры целые числа a, b. Для простоты фиксируйте максимальное количество цифр в числе (#define LENGTH 10) и заведите массивы A[], B[] в функции main().

- Напишите функцию Fill, которая принимает на вход число, массив и размер этого массива. Функция должна разбивать число на цифры и складывать их в данный массив, контролируя переполнение. Примените её к введённым числам и подготовленным массивам.

```
// 127 -> {7, 2, 1, 0, 0, ..., 0}
void Fill(int number, int digits[], int length);
```

- Напишите функцию Print, которая принимает на вход массив и его размер. Функция должна печатать цифры числа в прямом порядке через пробел. Не выводите ведущие нули, если сможете. Примените её к заполненным массивам.

```
// {7, 2, 1, 0, 0, ..., 0} -> "1 2 7"
void Print(int digits[], int length);
```

2. **Сложите с переносом (+1 балл).** Заведите массив C[] и целую переменную c.

- Напишите функцию Add, которая принимает на вход три массива и их размеры, складывает цифры из первых двух, помещая результат в третий. Примените к массивам, напечатайте результат.

3 → `void Add(int x[], int xLength, int y[], int yLength, int xy[], int xyLength);`

- 2 → `void Add(int x[], int xLength, int y[], int yLength, int xy[], int xyLength);`
- Напишите функцию Collect, которая принимает на вход массив и его размер, а возвращает собранное из цифр число. Соберите из цифр число c', сравните с правильным ответом c".

```
// {7, 2, 1, 0, 0, ..., 0} -> 127
int Collect(int digits[], int length);
```


3. **Выведите красиво (+1 балл).** Пример на сложение в столбик следует записывать в столбик, цифра под цифрой. Для этого нам придётся запоминать количество значащих цифр в каждом из чисел a , b , c и выбирать максимальное для выравнивания.
- Исправьте функции `Fill` и `Add` так, чтобы они возвращали количество значащих цифр в результате. В функцию `Add` у слагаемых теперь следует передавать не размер массива, а количество значащих цифр.
 - Передавайте в функцию `Print` не размер массива, а количество значащих цифр. Добавьте ещё один параметр — ширину для выравнивания, передавая туда максимальную из длин задействованных трёх чисел. Измените функцию так, чтобы она печатала ведущие пробелы, тем самым выравнивая числа по последней цифре. Не забудьте напечатать горизонтальную черту нужной длины (из знаков минус).
4. **(*) Обобщите алгоритм (+1 бонус).** Хотя числа мы вводим с клавиатуры всегда в десятичной системе (`%i`), никто не мешает нам внутренние вычисления проводить в произвольной системе счисления.
- Заведите константу `#define BASE 10`, определяющую систему счисления в алгоритме сложения в столбик. Поправьте функции `Fill`, `Add` и `Collect` так, чтобы они работали с массивами цифр в системе счисления, заданной в `BASE`.
 - Не забудьте поправить функцию `Print` для корректной печати цифр в системах с основанием больше 10. Проверьте правильность работы в разных системах счисления. Обратите внимание, что длина максимального числа изменяется (как логарифм по соответствующему основанию).
 - **Внимание!** Человек вводит числа a и b по-прежнему в десятичной системе, ждёт результат c в ней же. Другая система счисления присутствует только в массивах как демонстрация того, что во всех системах алгоритм одинаковый и приводит к одному и тому же результату.