**27.11 23**

// **Учебный пример.** Приведены отдельные функции.

// Функция **sortArray() ---** cортировка массива **mas(n)** не эффективная.

// Функция **sortArrayEffic()** --- сортировка массива **mas(n)** эффективная

// (без лишних просмотров массива)

// Функция **addElem()** вставки элемента в упорядоченный массив

// Функция **firstDigit()** --- нахождение старшей цифры натурального числа **n**

void **sortArray**(int mas[], int n) { // функция **sortArray()**

int x; // не эффект. сортировка

for (int i = 1; i < n; i++)

for (int j = 0; j < n - i; j++)

if (mas[j] > mas[j + 1])

{

x = mas[j]; mas[j] = mas[j + 1]; mas[j + 1] = x;

}

}

void **sortArrayEffic**(int mas[], int n) { // функция **sortArrayEffic()**

int flag = 1; int rab, i = 1; // эффект. сортировка

while (flag){

flag = 0;

for (int j = 0; j < n - i; j++)

if (mas[j] > mas[j + 1]){

rab = mas[j]; mas[j] = mas[j + 1]; mas[j + 1] = rab;

flag = 1;

}

i++;

}

}

// Функция **addElem()** вставки элемента в упорядоченный массив

void **addElem**(int arr[], int arrSize, int newElem){

int newElemIndex = arrSize;

for (int i = 0; i < arrSize; i++){ // поиск, куда вставить элемент

if (newElem < arr[i])

{

newElemIndex = i;

break;

}

}

for (int i = arrSize; i > newElemIndex; i--) // освобождаем место

arr[i] = arr[i - 1]; // для вставки элемента

arr[newElemIndex] = newElem; // вставка элемента

}

// Функция **olderDigit()** нахождения старшей цифры числа

int **olderDigit**(int num){

while (num > 9)

num = num / 10;

return num;

}