

Основы программирования

Лабораторная работа №8

Базовые алгоритмы обработки
одномерных массивов

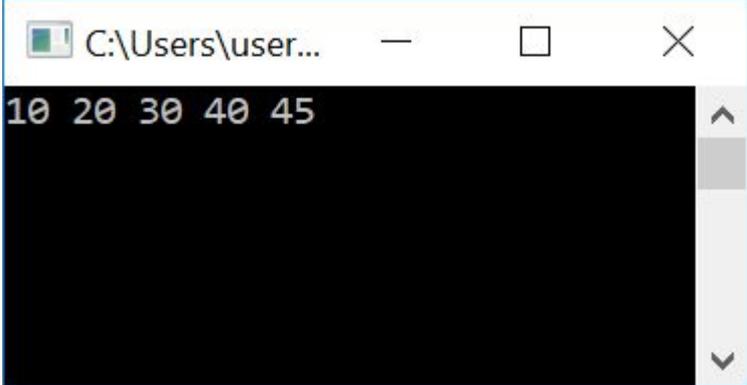
Власенко О.Ф.

Инициализация массива. Вывод в консоль

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    int a[5] = {10, 20, 30, 40, 45};
    int i = 0;
    while (i < 5) {
        printf("%d ", a[i]);
        i++;
    }

    {
        int x;
        scanf("%d", &x);
    }
}
```



A screenshot of a Windows console window. The title bar shows the path 'C:\Users\user...'. The console output displays the numbers '10 20 30 40 45' on a single line. The background of the console is black, and the text is white. There are standard window control buttons (minimize, maximize, close) in the title bar.

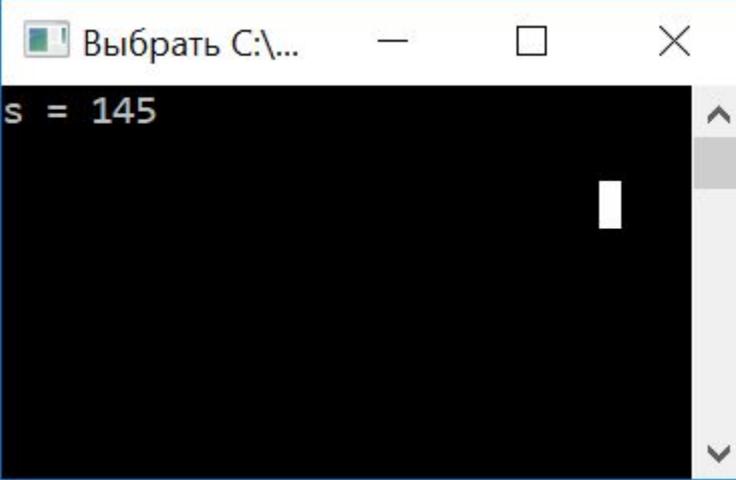
Подсчет суммы всех элементов

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    int a[5] = {10, 20, 30, 40, 45};
    int s = 0;
    int i = 0;
    while (i < 5) {
        s += a[i];
        i++;
    }

    printf("s = %d", s);

    {
        int x;
        scanf("%d", &x);
    }
}
```



The screenshot shows a Windows command prompt window with the title bar text "Выбрать C:\...". The window content displays the output of the program: "s = 145". The prompt is currently on a new line, indicating that the program has finished execution and is waiting for user input.

Ввод элементов через консоль. Подсчет количества четных элементов.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

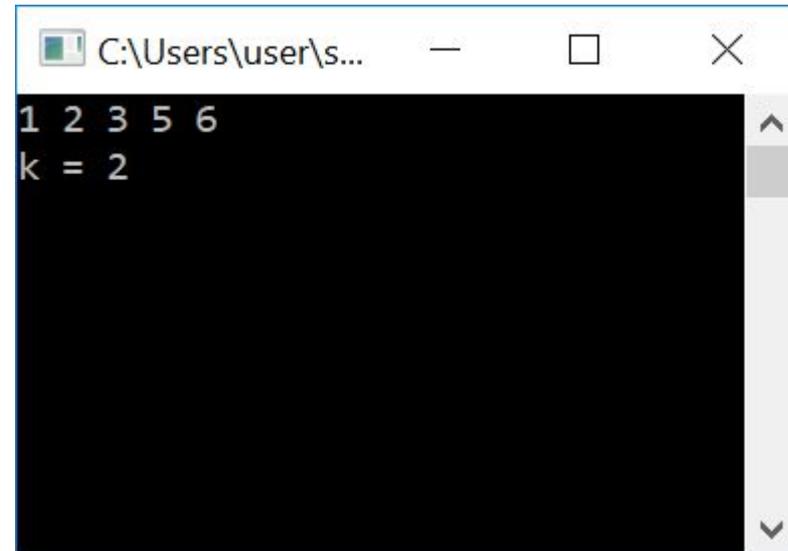
void main()
{
    int a[5];
    int k; // количество четных элементов
    int i;

    i = 0;
    while (i < 5) {
        scanf("%d", &a[i]);
        i++;
    }

    k = 0;
    i = 0;
    while (i < 5) {
        if (a[i] % 2 == 0) k++; // нашли четный элемент
        i++;
    }

    printf("k = %d", k);

    {
        int x;
        scanf("%d", &x);
    }
}
```

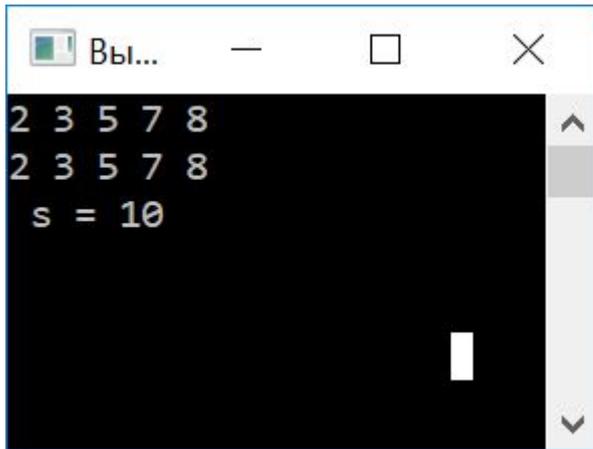


```
C:\Users\user\s...
1 2 3 5 6
k = 2
```

Задача 1

Ввести массив с клавиатуры.

Вывести получившийся массив и сумму четных элементов



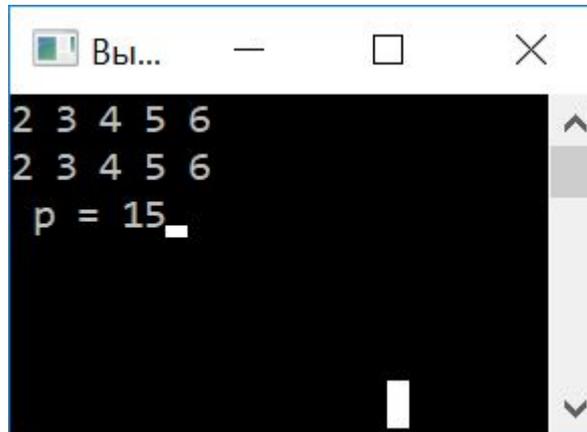
```
Вы...  —  □  ×  
2 3 5 7 8  
2 3 5 7 8  
s = 10
```

The screenshot shows a standard Windows command prompt window with a black background and white text. The window title bar contains the text 'Вы...' followed by standard minimize, maximize, and close buttons. The command prompt displays three lines of text: '2 3 5 7 8', '2 3 5 7 8', and 's = 10'. A white cursor is visible at the end of the third line.

Задача 2

Ввести массив с клавиатуры.

Вывести получившийся массив и произведение нечетных элементов



```
Вы... - □ ×
2 3 4 5 6
2 3 4 5 6
p = 15_
```

The screenshot shows a console window with a black background and white text. The window title is "Вы...". The text displayed is: "2 3 4 5 6", "2 3 4 5 6", and "p = 15_". A white cursor is positioned at the end of the last line. The window has standard Windows window controls (minimize, maximize, close) and a scrollbar on the right side.

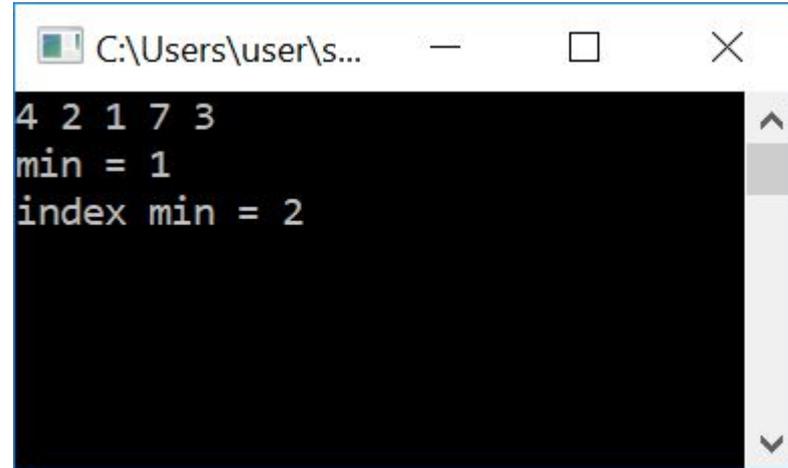
Поиск минимального элемента

```
int i;
int a[] = {4, 2, 1, 7, 3};

i = 0;
while (i < 5) {
    printf("%d ", a[i]);
    i++;
}
printf("\n");

int min = a[0];
int iMin = 0;
i = 1;
while (i < 5) {
    if (a[i] < min) {
        min = a[i];
        iMin = i;
    }
    i++;
}

printf("min = %d\n", min);
printf("index min = %d\n", iMin);
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\user\s...'. The window contains the following output:

```
4 2 1 7 3
min = 1
index min = 2
```

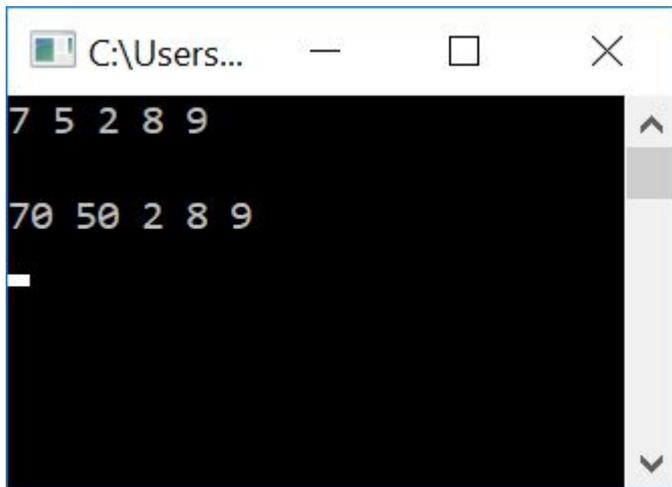
Задача 3

Ввести массив с клавиатуры (5 элементов).

Найти минимальный элемент.

Все элементы левее минимального увеличить в 10 раз.

Вывести получившийся массив.



```
C:\Users...  
7 5 2 8 9  
70 50 2 8 9  
_
```

```
int min = a[0];  
int iMin = 0;  
i = 1;  
while (i < 5) {  
    if (a[i] < min) {  
        min = a[i];  
        iMin = i;  
    }  
    i++;  
}  
  
i = 0;  
while (i < iMin) {  
    a[i] *= 10;  
    i++;  
}
```

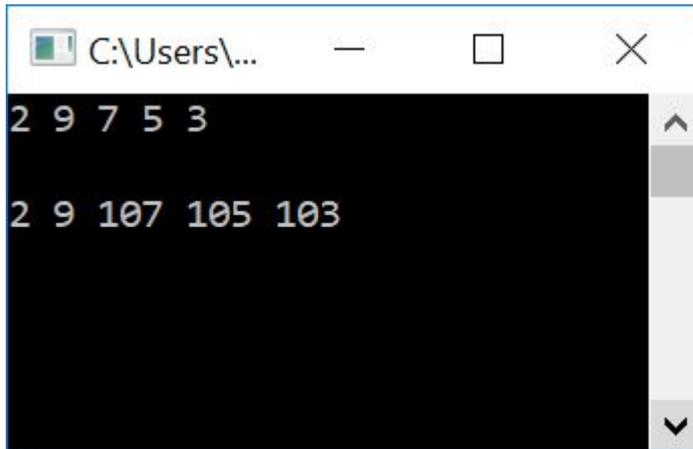
Задача 4

Ввести массив с клавиатуры (5 элементов).

Найти максимальный элемент.

Все элементы правее максимального увеличить на 100.

Вывести получившийся массив.

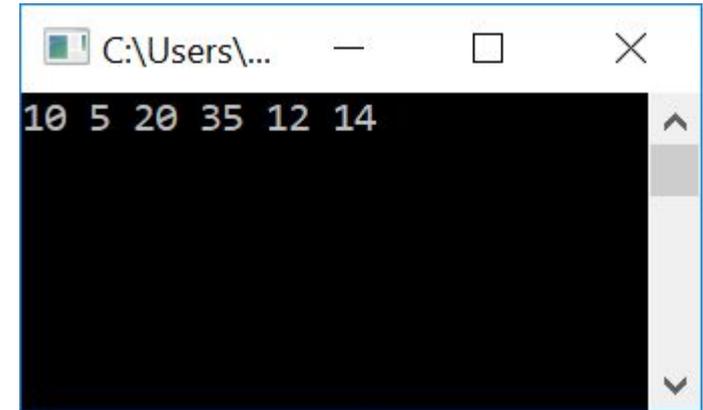


```
C:\Users\...  
2 9 7 5 3  
2 9 107 105 103
```

Массив с «переменной длиной»

Инициализация и вывод

```
void main() {  
    int i;  
    int a[10] = { 10, 5, 20, 35, 12, 14 };  
    int n = 6;  
  
    i = 0;  
    while (i < n) {  
        printf("%d ", a[i]);  
        i++;  
    }  
    printf("\n");  
}
```



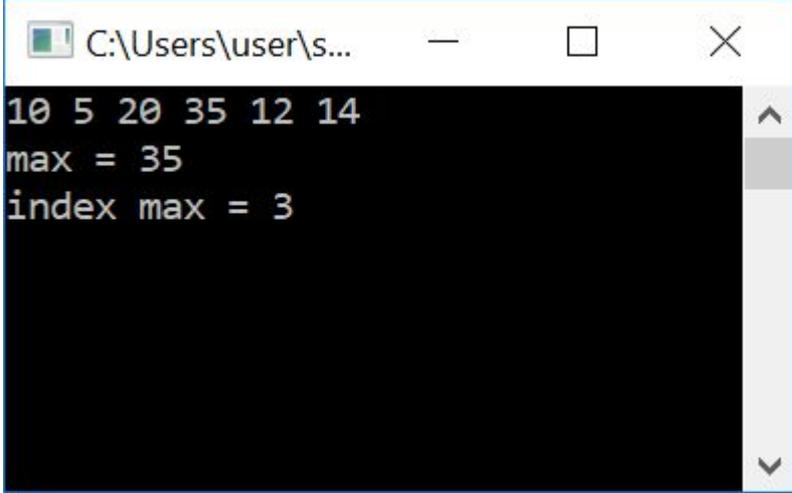
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\...' and standard window controls (minimize, maximize, close). The command prompt displays the output of the program: '10 5 20 35 12 14'.

Массив с «переменной длиной»

Поиск максимального

```
int max = a[0];
int iMax = 0;
i = 1;
while (i < n) {
    if (a[i] > max) {
        max = a[i];
        iMax = i;
    }
    i++;
}

printf("max = %d\n", max);
printf("index max = %d\n", iMax);
```



```
C:\Users\user\s...
10 5 20 35 12 14
max = 35
index max = 3
```

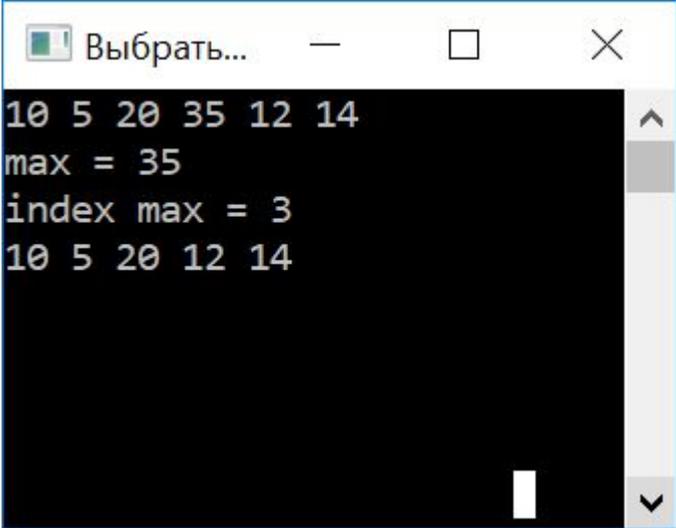
Массив с «переменной длиной»

Удаление максимального и вывод после удаления

```
i = iMax;
while (i < n - 1) {
    a[i] = a[i + 1];
    i++;
}
n--;

i = 0;
while (i < n) {
    printf("%d ", a[i]);
    i++;
}
printf("\n");

{
    int x;
    scanf("%d", &x);
}
```



```
Выбрать...
10 5 20 35 12 14
max = 35
index max = 3
10 5 20 12 14
```

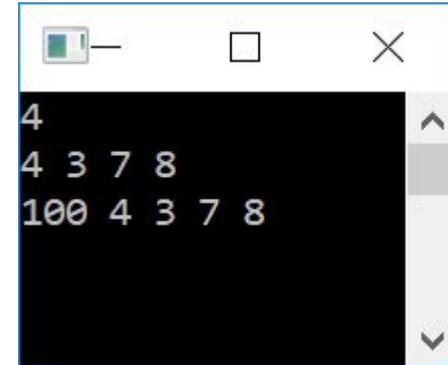
Массив с «переменной длиной»

Добавление элемента «100» в начало массива

```
void main() {
    int i;
    int a[10];
    int n;

    // ВВОД МАССИВА
    scanf("%d", &n);
    i = 0;
    while (i < n) {
        scanf("%d", &a[i]);
        i++;
    }

    // сдвиг массива и освобождение a[0]
    i = n;
    while (i > 0) {
        a[i] = a[i - 1];
        i--;
    }
    n++;
    // присвоение a[0] = 100
    a[0] = 100;
}
```



```
4
4 3 7 8
100 4 3 7 8
```

```
// ВЫВОД МАССИВА
i = 0;
while (i < n) {
    printf("%d ", a[i]);
    i++;
}
printf("\n");

{
    int x;
    scanf("%d", &x);
}
```

Задача 5

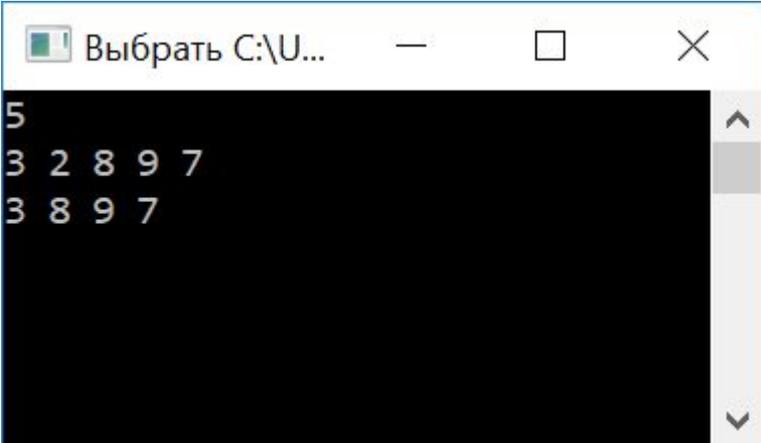
Ввести массив с клавиатуры (n элементов, $n \leq 10$).

Найти минимальный элемент.

Удалить минимальный элемент.

Вывести получившийся массив

```
void main() {  
    int i;  
    int a[10];  
    int n;  
  
    scanf("%d", &n);  
    i = 0;  
    while (i < n) {  
        scanf("%d", &a[i]);  
        i++;  
    }  
  
    int min = a[0];  
    int iMin = 0;  
    i = 1;  
    while (i < n) {  
        if (a[i] < min) {
```



The screenshot shows a Windows file explorer window with the title bar 'Выбрать C:\U...'. The main area displays a directory structure with three entries: a file named '5', a folder named '3 2 8 9 7', and another folder named '3 8 9 7'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Задача 6*

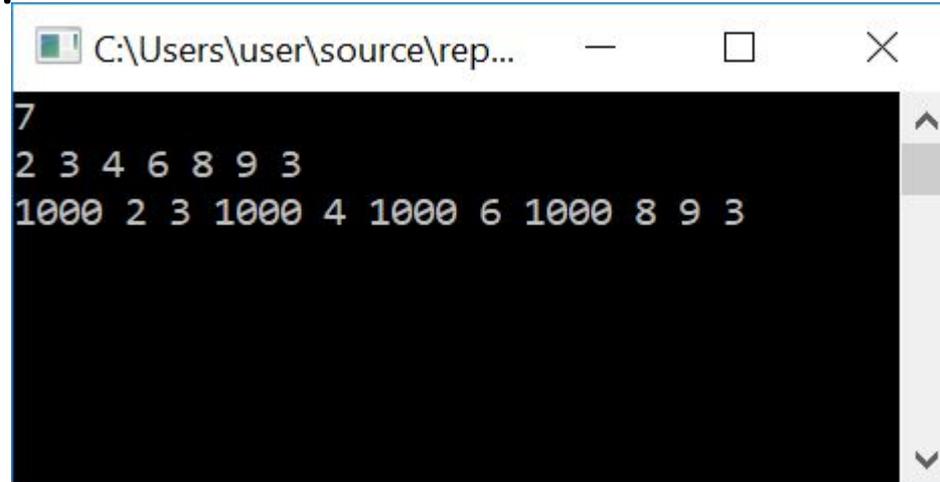
Ввести массив из N элементов ($N \leq 10$)

Перед каждым четным элементов вставить элемент со значением «1000».

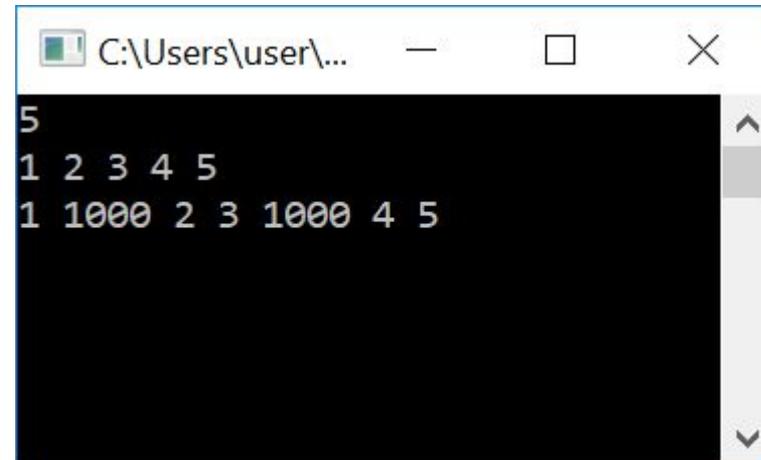
Вывести получившийся массив.

```
// ВВОД МАССИВА
scanf("%d", &n);
i = 0;
while (i < n) {
    scanf("%d", &a[i]);
    i++;
}

j = n - 1;
while (j >= 0) {
    if (a[j] % 2 == 0) {
        // сдвиг массива и освобождение a[j]
        i = n;
        while (i > i) {
```



```
C:\Users\user\source\rep...
7
2 3 4 6 8 9 3
1000 2 3 1000 4 1000 6 1000 8 9 3
```



```
C:\Users\user\...
5
1 2 3 4 5
1 1000 2 3 1000 4 5
```

Домашнее задание

Домашняя работа по лабораторной работе №8 включает в себя

- 1) Решение задачи по варианту 1 (выбираете задачу согласно последней цифре вашего номера по списку группы) или по варианту 2 – выбираете одну задачу по своему собственному вкусу.
- 2) Отчет

Отчет включает – если выбрали вариант 1

- a) Распечатку программы из домашней работы ПЛЮС блоксхема ПЛЮС трассировка для $n=5$
- b) Распечатку задач 1-5 из классной работы ПЛЮС блоксхема к каждой задаче

Отчет включает – если выбрали вариант 2

- c) Распечатку программы из домашней работы ПЛЮС блоксхема ПЛЮС трассировка для $n=8$
- d) Распечатку задач 1-6 из классной работы БЕЗ блоксхем

Домашнее задание – вариант 1

Вариант В1

Из массива удалить первый из четных элементов.

Пример: из массива A[5]: 1 3 4 5 6 должен

получиться

массив A[4]: 1 3 5 6

Вариант В2

Из массива удалить последний из четных элементов.

Пример: из массива A[5]: 1 3 4 5 6 должен

получиться

массив A[4]: 1 3 4 5

Вариант В3

Из массива удалить последний из нечетных элементов.

Пример: из массива A[5]: 1 3 4 5 6 должен

получиться

массив A[4]: 1 3 4 6

Вариант В4

Из массива удалить первый из нечетных элементов.

Пример: из массива A[5]: 1 3 4 5 6 должен

получиться

массив A[4]: 3 4 5 6

Вариант В5

После максимального из четных элементов вставить 0.

Пример: из массива A[5]: 1 3 4 5 6 должен

Вариант В6

После первого четного элемента вставить 0.

Пример: из массива A[5]: 1 6 8 3 4 должен получиться

массив A[6]: 1 6 0 8 3 4

Вариант В7

После последнего нечетного элемента вставить 0.

Пример: из массива A[5]: 1 3 8 3 5 должен получиться

массив A[6]: 1 3 8 3 5 0

Вариант В8

Удалить максимальный из четных элементов.

Пример: из массива A[5]: 2 3 4 7 5 должен получиться

массив A[4]: 2 3 7 5

Вариант В9

Удалить максимальный из кратных трем элементов.

Пример: из массива A[5]: 2 3 4 7 5 должен получиться

массив A[4]: 2 4 7 5

Вариант В10

После последнего кратного четырем элемента вставить 0.

Пример: из массива A[5]: 1 3 8 3 4 должен получиться

массив A[6]: 1 3 8 3 4 0

Домашнее задание – вариант 2

A1. Вводится массив A. При добавлении каждый новый элемент вставляется в массив таким образом, чтобы весь массив был упорядочен по возрастанию.

Пример: исходное состояние массив A - пустой

ввод 3 массив A 3

ввод 5 массив A 3 5

ввод 2 массив A 2 3 5

ввод 7 массив A 2 3 5 7

ввод 4 массив A 2 3 4 5 7

A2. Из массива удалить элементы, встречающиеся в массиве только один раз.

Пример: из массива A[7]: 9 1 4 9 1 9 0 должен

получиться массив A[5]: 9 1 9 1 9

A3. Из массива удалить самую длинную цепочку четных элементов.

Пример: из массива A[8]: 4 1 4 2 1 2 4 6 должен

получиться массив A[5]: 4 1 4 2 1

(самая длинная цепочка четных чисел включает элементы с 6 по 8: 2 4 6)

A4. Из массива A удалить те цепочки нечетных элементов, в которых нет ни одного элемента из массива B.

Пример: массив A[10]: 3 2 7 5 2 1 2 6 3 9

массив B[5]: 1 2 5 4 8

Массив A после удаления примет вид:

A[7]: 2 7 5 2 1 2 6

A5. Между массивами A и B обменять их самые длинные

цепочки из четных элементов.

Пример: массив A[10]: 3 2 4 6 2 1 1 1 8 9

массив B[7]: 1 0 5 5 4 3 3

В массиве A самая длинная цепочка:

2 4 6 2 (элементы со 2 по 5)

В массиве B самая длинная цепочка:

0 (элемент 2)

Массив A после перестановки в него цепочки из массива

B: A[7]: 3 0 1 1 1 8 9

Массив B после перестановки в него цепочки из массива

A: B[10]: 1 2 4 6 2 5 5 4 3 3

A6. Вводятся массивы A, B и C. Из массива A удалить элементы, совпадающие по значению с теми из элементов массива B, которых нет в массиве C.

Пример: массив A 7 3 7 5 4 9

массив B 1 3 4 2 3 6 4 1 3 5

массив C 3 1 8 9

из массива A не удаляются:

7 - нет в B

3 - есть в B, но есть и в C

9 - нет в B

из массива A удаляются:

5 - есть в B и нет в C

4 - есть в B и нет в C

массив A после обработки 7 3 7 9