

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Белгородский государственный технологический университет им.
В.Г. Шухова»**

(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №9

По дисциплине: «Основы программирования»

**Тема: «Использование функций при решении задач на одномерные
массивы»**

Выполнил: студент группы ВТ-231

Борченко Александр Сергеевич

Проверили:

Черников Сергей Викторович

Новожен Никита Викторович

Белгород 2023

Цель работы: получение навыков решения задач на одномерные массивы.

Содержание работы:

Задача 1:	3
Задача 2:	4
Задача 3:	6
Задача 4:	8
Задача 5:	10
Задача 6:	12
Задача 7:	14
Задача 8:	16
Задача 9:	18
Задача 10:	20

Задача 1: Если возможно, то упорядочить данный массив размера n по убыванию, иначе массив оставить без изменения.

Спецификация функции SortByNonDecreasing:

1. Заголовок: `void SortByNonDecreasing(int *const a, const size_t n)`
2. Назначение: Упорядочить массив a размером n по неубыванию, если это возможно, иначе – оставить массив без изменений.

Код задачи:

```
void SortByNonDecreasing(int *const a, const size_t n) {
    int can_sorted_arr = 1;
    for (size_t i = 0; i < n - 1; ++i) {
        for (size_t g = i + 1; g < n; ++g) {
            if (a[i] == a[g]) {
                can_sorted_arr = 0;
            }
        }
    }
    if (can_sorted_arr) {
        selectionSort(a, n);
    }
    OutputArray(a, n);
}
```

Тестовые данные:

Входные данные	Выходные данные
1 2 4	4 2 1
4 2 4	4 2 4
1 3 1 4	1 3 1 4
4 2 3 1	4 3 2 1

Задача 2: Дана целочисленная последовательность. Упорядочить по неубыванию часть последовательности, заключенную между первым вхождением максимального значения и последним вхождением минимального.

Код задачи:

```
#include <stdio.h>

void swap(int * const a, int * const b) { //обмен значениями
    const int temp = *a;
    *a = *b;
    *b = temp;
}

void InputArray(int *a, const int n) { //ввод массива a размером n
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

void OutputArray(int *a, const size_t n) { //вывод массива a размером n
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

// Функция для нахождения первого вхождения максимального значения
int findFirstMax(const int *a, const size_t n) {
    int max = a[0];
    int max_index = 0;

    for (int i = 1; i < n; i++) {
        if (a[i] > max) {
            max = a[i];
            max_index = i;
        }
    }

    return max_index;
}

// Функция для нахождения последнего вхождения минимального значения
int findLastMin(const int *a, const size_t n) {
    int min = a[0];
    int min_index = 0;

    for (int i = 1; i < n; i++) {
        if (a[i] <= min) {
            min = a[i];
            min_index = i;
        }
    }

    return min_index;
}

//Функция для сортировки массива по возрастанию
void selectionSort(int *a, const int size) {
    for (int i = 0; i < size - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < size; j++)
            if (a[j] < a[minPos])
                minPos = j;
    }
}
```

```

        swap(&a[i], &a[minPos]);
    }
}

int main()
{
    size_t n = 5;
    int a[n];

    InputArray(a, n);

    size_t max = findFirstMax(a, n);
    size_t min = findLastMin(a, n);

    int left = min < max ? min : max;
    int right = min > max ? min : max;
    selectionSort(a + left + 1, right - left - 1);

    OutputArray(a, n);

    return 0;
}

```

Тестовые данные:

Входные данные	Выходные данные
10 3 2 1 0	10 1 2 3 0
0 3 2 1 10	0 1 2 3 10

Задача 3: Если данная последовательность не упорядочена ни по неубыванию, ни по невозрастанию, найти среднее геометрическое¹⁷ положительных членов.

Код задачи:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

void InputArray(int *a, const int n) { //ввод массива a размером n
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

void OutputArray(int *a, const size_t n) { //вывод массива a размером n
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

//отсортировано по неубыванию
int SortedUnDecreasing(int *a, size_t n) {
    for(size_t i = 1; i < n; i++) {
        if(a[i] < a[i - 1]) {
            return 0;
        }
    }
    return 1;
}

//отсортировано по невозрастанию
int SortedUnIncreasing(int *a, size_t n) {
    for(size_t i = 1; i < n; i++) {
        if(a[i] > a[i - 1]) {
            return 0;
        }
    }
    return 1;
}

//Возвращает среднее геометрическое n чисел
double GeometricMeanArrayNumbers(const int *a, const size_t n){
    double sum = 1;
    double count = 0;
    for (size_t i = 0; i < n; i++){
        if (a[i] > 0) {
            sum *= a[i];
            count++;
        }
    }
    if (count == 0)
        sum = 0;

    return pow(sum, (1 / count));
}

int main()
{
    SetConsoleOutputCP(CP_UTF8);

    size_t n;
    scanf("%lld", &n);

    int a[n];
```

```
InputArray(a, n);

if (SortedUnDecreasing(a, n) || SortedUnIncreasing(a, n))
    printf("Последовательность упорядочена");
else
    printf("%f", GeometricMeanArrayNumbers(a, n));

return 0;
}
```

Тестовые данные:

Входные данные	Выходные данные
4 1 2	2
2 4 3	2.884499
1 2 4	Последовательность упорядочена
-1 -1 -1	Последовательность упорядочена

Задача 4: Если число x встречается в данной целочисленной последовательности, то упорядочить по неубыванию часть последовательности после первого вхождения x .

Код задачи:

```
#include <stdio.h>

//Ввод массива a размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Вывод массива a размера n.
void outputArray(const int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

//обмен двух значений
void swap(int *a, int *b) {
    float temp = *a;
    *a = *b;
    *b = temp;
}

//Возвращает индекс элемента x в массиве
int getIndexXElement(const int *a, const size_t n, const int x) {
    for (int i = 0; i < n; i++){
        if (a[i] == x)
            return i;
    }
    return -1;
}

//Сортирует массив по возрастанию
void selectionSort(int *a, const int size) {
    for (int i = 0; i < size - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < size; j++)
            if (a[j] < a[minPos])
                minPos = j;
        swap(&a[i], &a[minPos]);
    }
}

//Сортировка массива после поиска ID X
void sortAfter(int *a, const int n, int x) {
    long long x_id = getIndexXElement(a, n, x);
    if (x_id > -1)
        selectionSort(a + 1 + x_id, n - 1 - x_id);
}

int main(){
    size_t n;
    scanf("%lld", &n);

    int a[n];
    inputArray(a, n);
```



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

sortAfter(a, n, x);

outputArray(a, n);

return 0;
}
```

Тестовые данные:

Входные данные	Выходные данные
16 8 4 2 1 X = 4	16 8 4 1 2
16 8 4 2 1 X = 16	16 1 2 4 8
16 8 4 2 1 X = 1	16 8 4 2 1
16 8 4 2 1 X = 9	16 8 4 2 1

Задача 5: Даны две последовательности. Получить упорядоченную по невозрастанию последовательность, состоящую из тех членов первой последовательности, которых нет во второй.

Код задачи:

```
#include <stdio.h>

//Ввод массива а размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Вывод массива а размера n.
void outputArray(const int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

//Меняет местами два значения
void swap(int *a, int *b) {
    float temp = *a;
    *a = *b ;
    *b = temp;
}

//Сортирует массив по возрастанию
void selectionSort(int *a, const int size) {
    for (int i = 0; i < size - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < size; j++)
            if (a[j] > a[minPos])
                minPos = j;
        swap(&a[i], &a[minPos]);
    }
}

//Новый массив с. Добавляем элементы массива а, которые не встречаются в массиве b
void newArrC(const int *a, const size_t n1, const int *b, const size_t n2,
int *c, size_t *n3){
    int count_b = 0;
    int count_c = 0;
    for (size_t i = 0; i < n1; i++){
        while (count_b < n2 - 1 && b[count_b] < a[i])
            count_b++;
        if(b[count_b] != a[i]){
            c[count_c] = a[i];
            count_c++;
        }
    }
    *n3 = count_c;
}

int main()
{
    size_t n1, n2;
    scanf("%lld %lld", &n1, &n2);
```

```

int a[n1], b[n2];
inputArray(a, n1);
selectionSort(a, n1);
inputArray(b, n2);
selectionSort(b, n2);

size_t n3 = n1;
int c[n3];
newArrC(a, n1, b, n2, c, &n3);

outputArray(c, n3);

return 0;
}

```

Тестовые данные:

Входные данные	Выходные данные
A = {1 2 4} B = {4}	C = {2 1}
A = {1 2 2 4} B = {3}	C = {4 2 2 1}

Задача 6: Дана целочисленная последовательность, содержащая как положительные, так и отрицательные числа. Упорядочить последовательность следующим образом: сначала идут отрицательные числа, упорядоченные по невозрастанию, потом положительные, упорядоченные по неубыванию.

Код задачи:

```
#include <stdio.h>

//Ввод массива a размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Вывод массива a размера n.
void outputArray(const int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

//Меняет местами два значения
void swap(int *a, int *b) {
    float temp = *a;
    *a = *b;
    *b = temp;
}

//Сортирует массив по возрастанию
void selectionSort(int *a, const int size) {
    for (int i = 0; i < size - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < size; j++)
            if (a[j] < a[minPos])
                minPos = j;
        swap(&a[i], &a[minPos]);
    }
}

// Изменение порядка элементов массива на обратный
void reverse(int *a, const size_t n) {
    for (size_t i = 0, j = n - 1; i < j; i++, j--)
        swap(&a[i], &a[j]);
}

//Возвращает индекс первого элемента равного 0, или больше 0
int getIndexFirstPositive(const int *a, const size_t n) {
    for (int i = 0; i < n; i++){
        if (a[i] >= 0)
            return i;
    }
    return -1;
}

int main()
{
    size_t n;
    scanf("%lld", &n);

    int a[n];
```

```

inputArray(a, n);

selectionSort(a, n);
int positive = getIndexFirstPositive(a,n);
if (positive > 0)
    reverse(a, getIndexFirstPositive(a, n));

outputArray(a, n);

return 0;
}

```

Тестовые данные:

Входные данные	Выходные данные
3 2 1 1 -4 -5 -6	-4 -5 -6 1 1 2 3
-3 -2 -1 0 1 2 3 4	-1 -2 -3 0 1 2 3 4
1 2 3 4 5	1 2 3 4 5

Задача 7: Дана целочисленная последовательность (по определению содержащая как положительные, так и отрицательные элементы) и целое число x . Определить, есть ли x среди членов последовательности, и если нет, то найти члены последовательности, ближайšie к x снизу и сверху.

Код задачи:

```
#include <stdio.h>
#include <windows.h>

//Ввод массива а размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Поиск чисел, ближайших к числу x, если числа x нет в массиве
int FindingNumbersClosestToNumberX(const int *a, const size_t n, const int x,
int *min, int *max){
    for (size_t i = 0; i < n; i++){
        if (a[i] > x){
            if (*max > a[i])
                *max = a[i];
        }
        else if (a[i] < x){
            if (*min < a[i])
                *min = a[i];
        }
        else
            return 1;
    }

    return 0;
}

int main(){
    SetConsoleOutputCP(CP_UTF8);

    size_t n;
    scanf("%lld ", &n);

    int a[n];
    inputArray(a, n);

    int x;
    scanf("%d", &x);

    int min = INT_MIN;
    int max = INT_MAX;

    if (FindingNumbersClosestToNumberX(a, n, x, &min, &max))
        printf("%d - элемент последовательности", x);
    else {
        if (min == INT_MIN)
            printf("-∞ ");
        else
            printf("%d ", min);

        if (max == INT_MAX)
            printf("∞");
        else
            printf("%d", max);
    }
}
```

```

    }

    return 0;
}

```

Тестовые данные:

Входные данные	Выходные данные
1 3 6 2 5 X = 0	$-\infty$ 1
1 3 6 2 5 X = 4	3 5
1 3 6 2 5 X = 6	x - элемент последовательности

Задача 8: Дана целочисленная последовательность. Получить массив из уникальных элементов последовательности.

Код задачи:

```
#include <stdio.h>
#include <windows.h>

int getItemPosition(const int *a, const int n, const int x) {
    for (int i = 0; i < n; i++) {
        if (a[i] == x) {
            return i;
        }
    }

    return -1;
}

//Ввод массива a размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Вывод массива a размера n.
void outputArray(const int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

void saveUniqueOnce(int *a, int n) {
    int b[n];
    int j = 0;
    for(size_t i = 0; i < n; i++) {
        int pos = getItemPosition(a, n, a[i]);
        if(getItemPosition(a+pos+1, n, a[i]) < 0) {
            b[j] = a[i];
            j++;
        }
    }
    if (j == 0)
        printf("Последовательность пуста");
    else
        outputArray(b, j);
}

int main()
{
    SetConsoleOutputCP(CP_UTF8);
    size_t n;
    scanf("%lld", &n);

    int a[n];
    inputArray(a, n);

    saveUniqueOnce(a, n);

    return 0;
}
```


Тестовые данные:

Входные данные	Выходные данные
1 2 4 1 2	4
1 2 3 4 5	1 2 3 4 5
1 1 1 1 1	Последовательность пуста

Задача 9: Определить, можно ли, переставив члены данной целочисленной последовательности длины n ($n > 1$), получить геометрическую прогрессию с знаменателем q ($|q| \neq 1$). Разрешимое допущение: знаменатель прогрессии – целое число.

Код задачи:

```
#include <stdio.h>
#include <math.h>

//Ввод массива а размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//обмен значениями
void swap(int *a, int *b) {
    float temp = *a;
    *a = *b;
    *b = temp;
}

//Сортировка массива по возрастанию модуля числа
void selectionSortAbsNumber(int *a, const int size) {
    for (int i = 0; i < size - 1; i++) {
        int minPos = i;
        for (int j = i + 1; j < size; j++)
            if (abs(a[j]) < abs(a[minPos]))
                minPos = j;
        swap(&a[i], &a[minPos]);
    }
}

//Проверяет является ли массив геометрической прогрессией
int isArrGeometricProgression(const int *a, const size_t n) {
    if (!a[0] || (a[1] / a[0]) == 1)
        return 0;

    int step = a[1] / a[0];
    for (size_t i = 2; i < n; i++) {
        if ((a[i] / a[i-1]) != step)
            return 0;
    }

    return 1;
}

int main()
{
    size_t n;
    scanf("%lld ", &n);

    int a[n];
    inputArray(a, n);
    selectionSortAbsNumber(a, n);

    if (isArrGeometricProgression(a, n))
        printf("Yes");
    else
        printf("No");
}
```

```
return 0;  
}
```

Тестовые данные:

Входные данные	Выходные данные
4 1 2	Yes
-1 -4 -16 2 8	Yes
0 0 0	No
1 1 1 1 -1	No

Задача 10: **Найти сумму четных цифр элементов массива из положительных чисел.

Код задачи:

```
#include <stdio.h>

//Ввод массива a размера n
void inputArray(int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

//Вывод массива a размера n.
void outputArray(const int *a, const size_t n) {
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

int getEvenSumDigits(long long x) {
    int sum = 0;
    while (x > 0) {
        if(x % 2 == 0) {
            sum += x % 10;
        }
        x /= 10;
    }
    return sum;
}

int sumEvenDigitsPositiveNumbers(int *a, size_t n) {
    int sum = 0;
    for(size_t i = 0; i < n; i++) {
        if(a[i] > 0) {
            sum += getEvenSumDigits(a[i]);
        }
    }
    return sum;
}

int main()
{
    size_t n;
    scanf("%lld", &n);

    int a[n];
    inputArray(a, n);
    printf("%d", sumEvenDigitsPositiveNumbers(a, n));

    return 0;
}
```

Тестовые данные:

Входные данные	Выходные данные
-2 -4 -6 -8 -10	0
1 2 12 14 101 -6	8

Вывод: в ходе выполнения лабораторной работы я получил навыки использования функций при решении задач на одномерные массивы.