

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 №4d
по дисциплине: Основы программирования
тема: «Рекурсивные функции»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович

Проверили:
Притчин Иван Сергеевич
Черников Сергей Викторович

Белгород 2021 г.

Цель работы: получение навыков написания рекурсивных функций.

Содержание отчета:

- Тема лабораторной работы.
- Цель лабораторной работы.
- Решения задач. Для каждой задачи указать:
 - Условие задачи.
 - Исходный код рекурсивных функций без спецификаций.
- Задачи с одной звездочкой **обязаны** содержать функцию-обёртку

Решение задач:

Задача №1. Определить количество цифр в тексте, вводимом с клавиатуры. Текст заканчивается символом перехода на новую строку \n.

Код программы:

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

int countDigit() {
    char a = getchar();
    if (a == '\n')
        return 0;
    return isdigit(a) + countDigit();
}

int main() {
    printf("%d", countDigit());
    return 0;
}
```

Задача №2. Вывести данное натуральное число в восьмеричной системе счисления.

Код программы:

```
#include <stdio.h>

void oct(int a) {
    if (a > 7) {
        int b = a >> 3;
        oct(b);
    }
    printf("%d", a & 7);
}

int main() {
    int a;
    scanf("%d", &a);

    oct(a);

    return 0;
}
```

Задача №3. Дан знаменатель и первый член геометрической прогрессии. Вычислить n -й член прогрессии.

Код программы:

```
#include <stdio.h>

int nGeomProgression_(int const q, int a, int const n, int i) {
    i++;
    if (i == n)
        return a;
    else if (i < n)
        return nGeomProgression_(q, a * q, n, i);
}

int nGeomProgression(int const q, int a, int const n) {
    return nGeomProgression_(q, a, n, 0);
}

int main() {
    int q, a, n;
    scanf("%d %d %d", &q, &a, &n);

    printf("%d", nGeomProgression(q, a, n));

    return 0;
}
```

Задача №4. Дан массив a размера n . Необходимо проверить, является ли он упорядоченным по неубыванию.

Код программы:

```
#include <stdio.h>

void inputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        scanf("%d", &a[i]);
}

int isNonDecreasing(int const *a, int n) {
    n--;
    if (n == 1 && a[0] <= a[1])
        return 1;
    else if (a[n-1] <= a[n])
        return isNonDecreasing(a, n);
    return 0;
}

int main() {
    int n;
    scanf("%d", &n);
    int a[n];
    inputArray(a, n);

    printf("%d", isNonDecreasing(a, n));

    return 0;
}
```

Задача №5. Дана упорядоченная по убыванию последовательность целых чисел. Определить, есть ли среди членов данной последовательности число x , и если есть, найти номер этого члена. Бинарным поиском не пользоваться.

Код программы:

```
#include <stdio.h>

void inputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        scanf("%d", &a[i]);
}

int isFindNumberX(int const *const a, int n, int const x) {
    n--;
    if (a[n] == x)
        return 1;
    else if (a[n] < x)
        return isFindNumberX(a, n, x);
    return 0;
}

int main() {
    int n, x;
    scanf("%d %d", &n, &x);
    int a[n];
    inputArray(a, n);

    printf("%d", isFindNumberX(a, n, x));

    return 0;
}
```

Задача №6. *Найти номер первого вхождения минимального значения в последовательность длины n (линейный поиск).

Код программы:

```
#include <stdio.h>

void inputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        scanf("%d", &a[i]);
}

void updateIndexMin(int const *const a, int n, int *minIndex, int *min) {
    if (a[n] < *min) {
        *min = a[n];
        *minIndex = n;
        return;
    }
}

void indexMin_(int const *const a, int n, int *minIndex, int *min) {
    n--;
    if (n < 0)
        return;
    updateIndexMin(a, n, minIndex, min);
    indexMin_(a, n, minIndex, min);
}

int indexMin(int const *const a, int const n) {
    int min = a[n - 1], minIndex = n - 1;
    indexMin_(a, n, &minIndex, &min);
    return minIndex;
}
```

```

int main() {
    int n;
    scanf("%d", &n);
    int a[n];
    inputArray(a, n);

    printf("%d", indexMin(a, n));

    return 0;
}

```

Задача №7. Даны натуральные числа **a** и **b**. Определить, могут ли эти числа быть соседними членами последовательности Фибоначчи.

Код программы:

```

#include <stdio.h>

void swap(long long *a, long long *b) {
    long long tmp = *a;
    *a = *b;
    *b = tmp;
}

int isFibonacci_(long long first, long long second,
                 long long const a, long long const b) {
    long long tmp = first + second;
    if (tmp == a + b && a == first && b == second)
        return 1;
    else if (tmp > a + b)
        return 0;
    return isFibonacci_(second, tmp, a, b);
}

int isFibonacci(long long a, long long b) {
    return isFibonacci_(1, 1, a, b);
}

int main() {
    long long a, b;
    scanf("%lld %lld", &a, &b);

    if (a > b)
        swap(&a, &b);

    if (isFibonacci(a, b))
        printf("YES");
    else
        printf("NO");

    return 0;
}

```

Задача №8. Вывести в обратном порядке символы данного текста, вводимого с клавиатуры, которые не являются цифрами. Текст заканчивается символом перехода на новую строку `\n`.

Код программы:

```
#include <stdio.h>

void reverseText() {
    char n = (char)getchar();
    if (n == '\n')
        return;
    else if (n >= '0' && n <= '9') {
        reverseText();
        return;
    }
    reverseText();
    printf("%c", n);
}

int main() {
    reverseText();

    return 0;
}
```

Задача №9. Дан n -й член арифметической прогрессии, ее разность и значение n . Вычислить первый член прогрессии.

Код программы:

```
#include <stdio.h>

int firstElementArithmeticSequence(int n, int const d, int an) {
    n--;
    if (n == 0)
        return an;
    firstElementArithmeticSequence(n, d, an - d);
}

int main() {
    int n, d, an;
    scanf("%d %d %d", &n, &d, &an);

    printf("%d", firstElementArithmeticSequence(n, d, an));

    return 0;
}
```

Задача №10. * С клавиатуры вводятся положительные вещественные числа a_1, a_2, \dots, a_n . Признак конца ввода – отрицательное число.

Код программы:

```
#include <stdio.h>

void floatAverage_(double last) {
    double a;
    scanf("%lf", &a);
    if (a < 0)
        return;
    floatAverage_(a);
    printf("%lf ", (last + a) / 2);
}

void floatAverage() {
    double a;
    scanf("%lf", &a);
    floatAverage_(a);
}
```

```
int main() {
    floatAverage();

    return 0;
}
```

Задача №11. Реализовать функцию `any`.

Код программы:

```
int any(int const *const a, int const n, int (*f)(int)) {
    if (n && f(a[n - 1]))
        return any(a, n - 1, *f);
    return n != 0;
}
```

Задача №12. Реализовать функцию `all`.

Код программы:

```
int all(int const *const a, int const n, int (*f)(int)) {
    if (n && f(a[n - 1]))
        return all(a, n - 1, *f);
    return n == 0;
}
```

Задача №13. Реализовать алгоритм бинарного поиска.

Код программы:

```
#include <stdio.h>

void inputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        scanf("%d", &a[i]);
}

int binSearch_(int const *const a, int const x, int left, int right) {
    if (left + 1 >= right)
        return left;
    int middle = left + (right - left) / 2;
    if (a[middle] > x)
        right = middle;
    else
        left = middle;
    return binSearch_(a, x, left, right);
}

int binSearch(int const *const a, int n, int const x) {
    return binSearch_(a, x, -1, n);
}

int main() {
    int n, x;
    scanf("%d %d", &n, &x);
    int a[n];
    inputArray(a, n);

    int index = binSearch(a, n, x);
    printf("%d", index);
    if (index == -1 || index == n || a[index] != x)
        printf("x does not belong to set a");
    else
        printf("The x value is the %dth element in array a", index);

    return 0;
}
```

Задача №14. Реализовать сортировку выбором.

Код программы:

```
#include <stdio.h>

void inputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        scanf("%d", &a[i]);
}

void outputArray(int *const a, int const size) {
    for (int i = 0; i < size; i++)
        printf("%d", a[i]);
}

void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int min(int const *const a, int n, int *maxIndex) {
    n--;
    if (n < 0)
        return *maxIndex;
    if (a[n] > a[*maxIndex])
        *maxIndex = n;
    return min(a, n, maxIndex);
}

void selectionSort(int *const a, int n) {
    n--;
    if (n < 0)
        return;
    int maxIndex = n;
    swap(&a[min(a, n, &maxIndex)], &a[maxIndex]);
    selectionSort(a, n);
}

int main() {
    int n;
    scanf("%d", &n);
    int a[n];
    inputArray(a, n);

    selectionSort(a, n);

    outputArray(a, n);

    return 0;
}
```