|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра технологий искусственного интеллекта |

ПРАКТИЧЕСКАЯ РАБОТА № 7

по дисциплине

«Процедурное программирование»

Обучающийся Сысоенко Глеб Максимович

Группа КАБО-01-23

Руководитель *Яковлев Д. А*

Москва 2023

**Практическая работа №7**

**Тема:** «Массивы как аргументы функций»

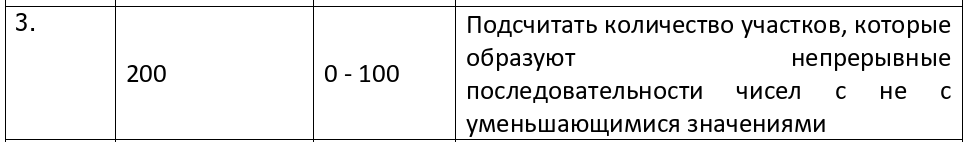
**Цель практической работы:**

Целью данной практической работы освоить на практике работу с функциями, получающими массивы в качестве аргументов.

**Описание:**

Выделите память для хранения массива чисел и заполните его любым способом, передать массив в функцию в качестве аргумента. Выполните задание по варианту

**Задание:**



**Код программы:**

// LAB 7

//Подсчитать количество участков, которые образуют

//непрерывные последовательности чисел с не с уменьшающимися значениями

//Размер массива 200, дипапаз значений 100+

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <synchapi.h>

int komunizm(int \*A){

int i, chitalka = 0, max\_ln = 0, c = 0;

for (i = 0; i < 200; i++) {

if (A[i] > A[i + 1] && A[i + 1] > max\_ln) {

c ++;

printf("A ITI + 1 %d NOMER -> %d| A ITI %d NOMER -> %d\n", A[i + 1], i+1, A[i], i);

} else {

if (c >= 1) {

max\_ln = A[i];

chitalka += 1;

c = 0;

printf("%d\n", max\_ln);

}

}

}

printf("\n| KOL-VO UD USL -> %d AND MAX ZN POSL POSLED -> %d |", chitalka, max\_ln);

}

int main() {

int \*A;

int i, chitalka = 0, max\_ln = 0, c = 0, result;

srand(time(NULL));

A = calloc(200, sizeof(int));

for (i = 0; i < 200; i++) {

A[i] = (rand() % 100);

printf(" |ZNACHENIE %d NOMER %d| ", A[i], i);

}

printf("\n\n");

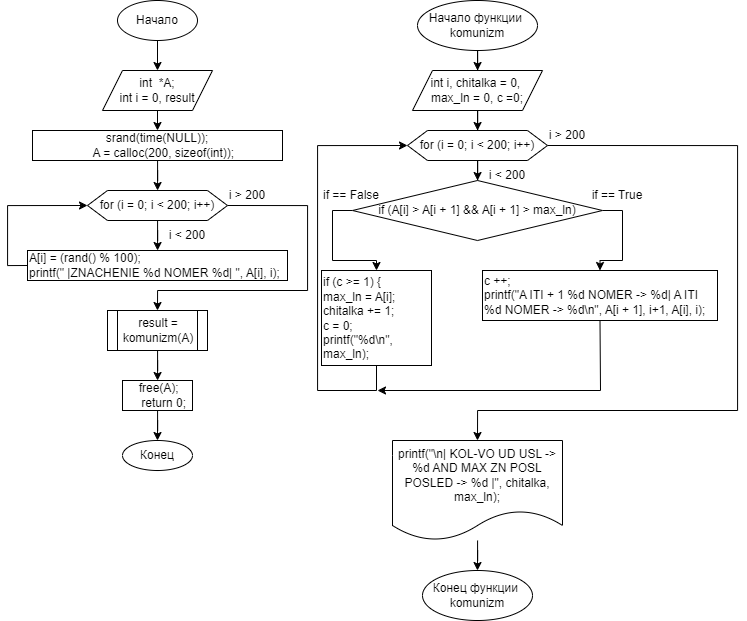
result = komunizm(A);

free(A);

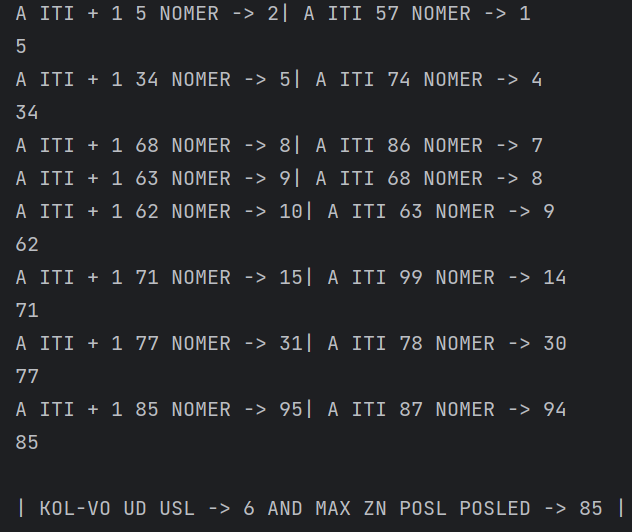
return 0;

}

**Блок-схема:**



**Скриншот выполнения программы:**



**Контрольные вопросы:**

**1.** Для передачи массива в качестве аргумента функции необходимо, что бы один из аргументов функции был указателем на массив.

**2.** Для организации динамического выделения памяти необходимо использовать функции malloc и calloc перед этим задать задать переменные, как указатели.

**3.** Для корректной работы программы, в противном случае может закончиться память на машине.

**4.** Malloc – просто выделяет участок памяти, а calloc ещё и зануляет все элементы.

**5.** Очистка памяти после использования динамического выделенного участка памяти(массив).

**6.** Когда надо можем выделить память по ходу процесса, можем изменять размерность и потом освободить эту память для оптимизации процессов.