Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЁТ

к лабораторной работе №5

на тему

ДИНАМИЧЕСКОЕ ВЫДЕЛЕНИЕ ПАМЯТИ

Вариант 15

Проверил Выполнил

асс. каф. ЭВМ студент гр. 318302 П.С. Жуковец Мизеркин Никита

МИНСК 2023

**1 ЗАДАНИЕ №1**

* 1. **Условие**

Заполнить массив случайными числами. Удалить из массива все элементы с чётным значением.

* 1. **Алгоритм решения**

Алгоритм программы выглядит следующим образом (рис. 1.1):

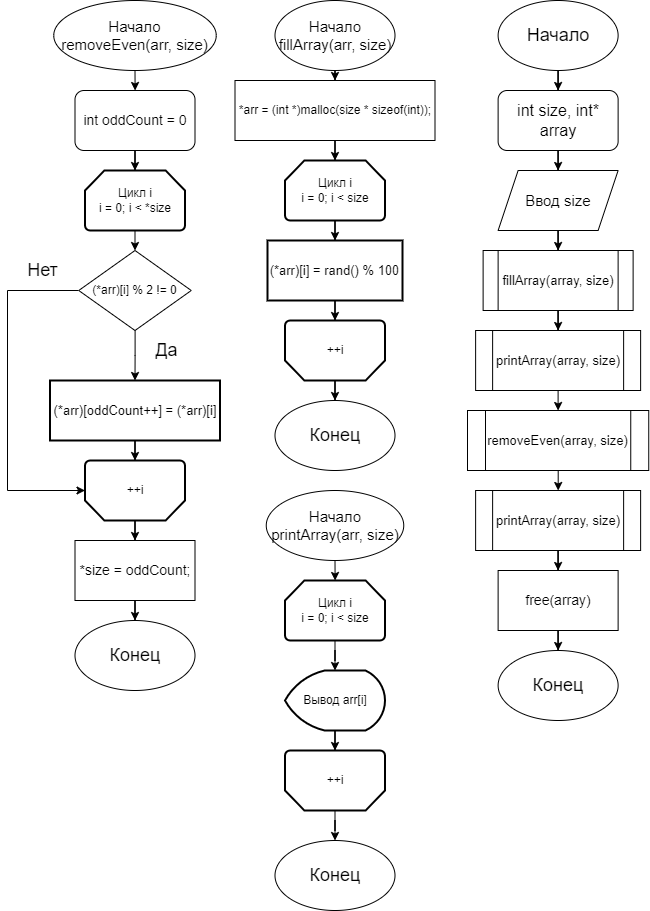


Рисунок 1.1 – Алгоритм задачи 1

* 1. **Листинг программы**

Таким образом выглядит код Си-программы, решающий задачу.

#include <windows.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void fillArray(int \*\*arr, int size) {

\*arr = (int \*)malloc(size \* sizeof(int));

srand(time(NULL));

for (int i = 0; i < size; ++i) {

(\*arr)[i] = rand() % 100;

}

}

void removeEven(int \*\*arr, int \*size) {

int oddCount = 0;

for (int i = 0; i < \*size; ++i) {

if ((\*arr)[i] % 2 != 0) {

(\*arr)[oddCount++] = (\*arr)[i];

}

}

\*size = oddCount;

void printArray(int \*arr, int size) {

for (int i = 0; i < size; ++i) {

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

SetConsoleOutputCP(CP\_UTF8);

int \*array;

int size;

printf("Введите размер массива: ");

scanf("%d", &size);

fillArray(&array, size);

printf("Исходный массив:\n");

printArray(array, size);

removeEven(&array, &size);

printf("Массив после удаления четных элементов:\n");

printArray(array, size);

free(array);

return 0;

}

**2 ЗАДАНИЕ №2**

**2.1 Условие**

В двумерном массиве натуральных чисел (количество чисел в строке может быть различным, последнее число - 0) удалить строку с минимальной суммой элементов (порядок остальных строк не менять).

**2.2 Алгоритм решения**

Алгоритм данной программы будет выглядеть следующим образом (рис. 2.1):

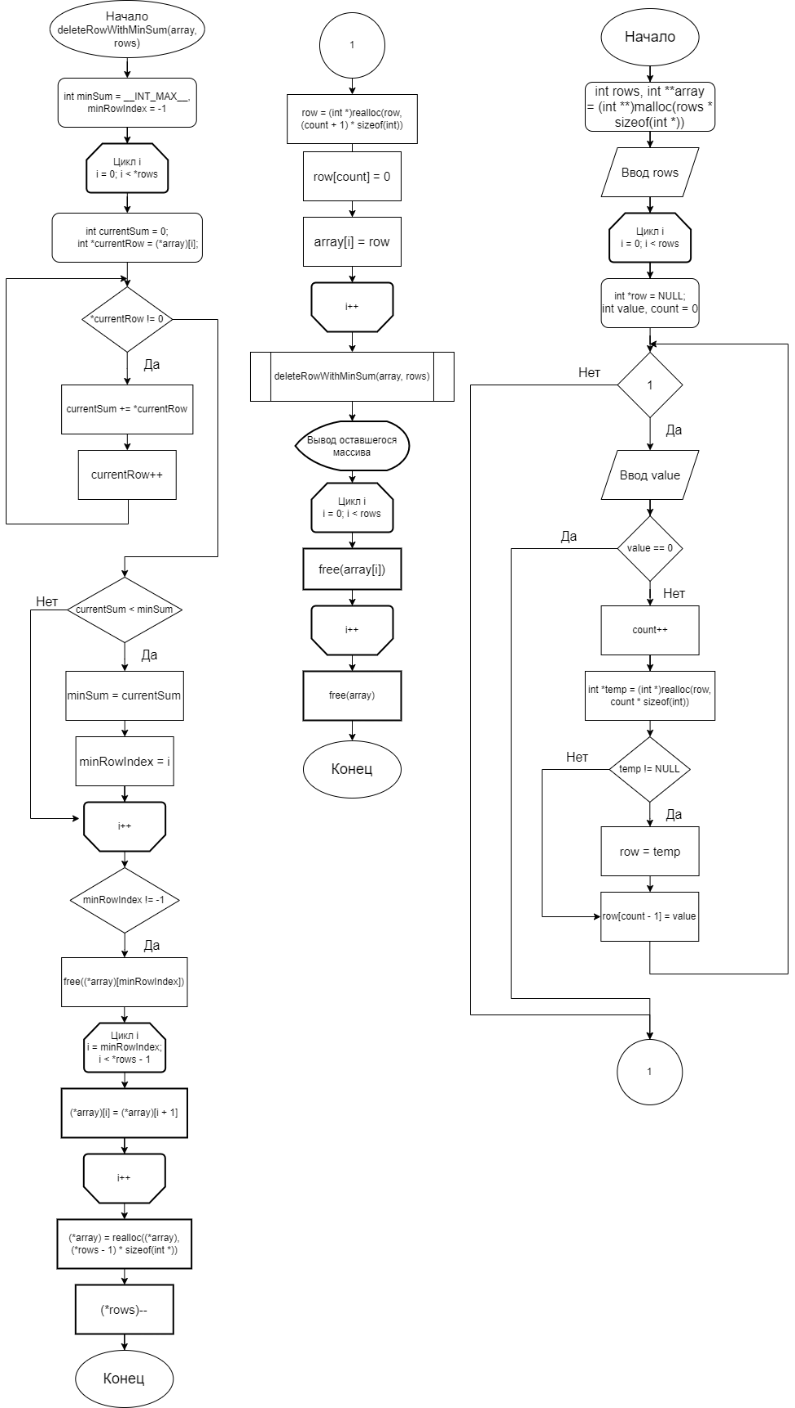


Рисунок 2.1 – Алгоритм задачи 2

**2.3 Листинг программы**

Код такой программы будет выглядеть так:

#include <windows.h>

#include <stdio.h>

#include <stdlib.h>

void deleteRowWithMinSum(int \*\*\*array, int \*rows) {

int minSum = \_\_INT\_MAX\_\_;

int minRowIndex = -1;

for (int i = 0; i < \*rows; i++) {

int currentSum = 0;

int \*currentRow = (\*array)[i];

while (\*currentRow != 0) {

currentSum += \*currentRow;

currentRow++;

}

if (currentSum < minSum) {

minSum = currentSum;

minRowIndex = i;

}

}

if (minRowIndex != -1) {

free((\*array)[minRowIndex]);

for (int i = minRowIndex; i < \*rows - 1; i++) {

(\*array)[i] = (\*array)[i + 1];

}

(\*array) = realloc((\*array), (\*rows - 1) \* sizeof(int \*));

(\*rows)--;

}

}

int main() {

SetConsoleOutputCP(CP\_UTF8);

int rows;

printf("Введите количество строк в массиве: ");

scanf("%d", &rows);

int \*\*array = (int \*\*)malloc(rows \* sizeof(int \*));

if (array == NULL) {

printf("Ошибка выделения памяти.\n");

return -1;

}

for (int i = 0; i < rows; i++) {

printf("Введите элементы строки %d (0 для завершения): ", i + 1);

int \*row = NULL;

int value, count = 0;

while (1) {

scanf("%d", &value);

if (value == 0) {

break;

}

count++;

int \*temp = (int \*)realloc(row, count \* sizeof(int));

if (temp != NULL) {

row = temp;

} else {

printf("Ошибка выделения памяти.\n");

exit(EXIT\_FAILURE);

}

row[count - 1] = value;

}

row = (int \*)realloc(row, (count + 1) \* sizeof(int));

row[count] = 0;

array[i] = row;

}

deleteRowWithMinSum(&array, &rows);

printf("Оставшийся массив:\n");

for (int i = 0; i < rows; i++) {

int \*currentRow = array[i];

while (\*currentRow != 0) {

printf("%d ", \*currentRow);

currentRow++;

}

printf("0\n");

}

for (int i = 0; i < rows; i++) {

free(array[i]);

}

free(array);

return 0;

}

**3 ЗАДАНИЕ №3**

* 1. **Условие**

Дан двумерный массив ненулевых целых чисел. Определить максимально длинную последовательность положительных чисел. Массив просматривается построчно сверху вниз, а в каждой строке - слева направо. Сохранение знака при переходе на новую строку также учитывать.

* 1. **Алгоритм решения**

Данная блок-схема характеризует алгоритм работы программы (рис.

3.1):

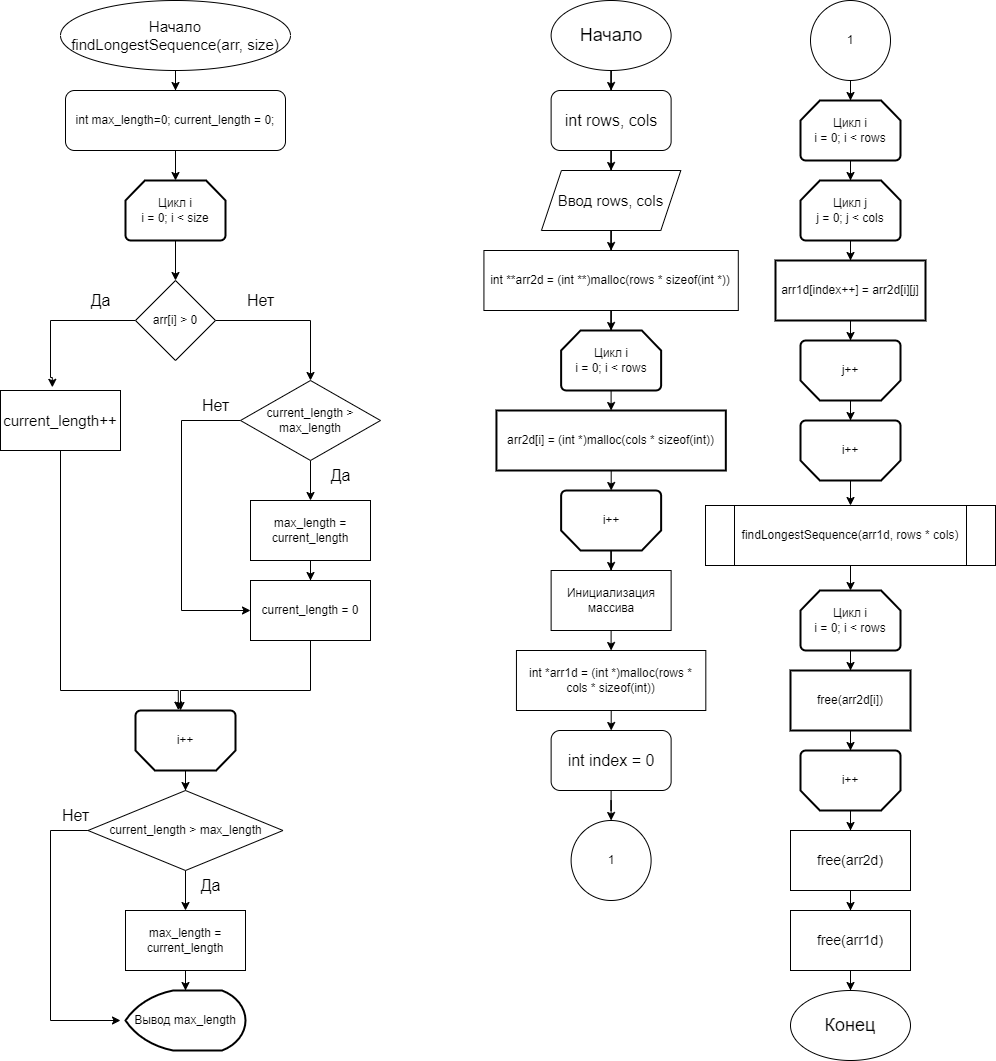


Рисунок 3.1 – Алгоритм задачи 3

* 1. **Листинг программы**

Код этой программы выглядит так:

#include <windows.h>

#include <stdio.h>

#include <stdlib.h>

void findLongestSequence(int \*arr, int size) {

int max\_length = 0;

int current\_length = 0;

for (int i = 0; i < size; i++) {

if (arr[i] > 0) {

current\_length++;

} else {

if (current\_length > max\_length) {

max\_length = current\_length;

}

current\_length = 0;

}

}

if (current\_length > max\_length) {

max\_length = current\_length;

}

printf("Максимальная длина последовательности положительных чисел: %d\n", max\_length);

}

int main() {

SetConsoleOutputCP(CP\_UTF8);

int rows, cols;

printf("Введите количество строк и столбцов: ");

scanf("%d %d", &rows, &cols);

int \*\*arr2d = (int \*\*)malloc(rows \* sizeof(int \*));

for (int i = 0; i < rows; i++) {

arr2d[i] = (int \*)malloc(cols \* sizeof(int));

}

printf("Введите элементы массива:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

scanf("%d", &arr2d[i][j]);

}

}

int \*arr1d = (int \*)malloc(rows \* cols \* sizeof(int));

int index = 0;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

arr1d[index++] = arr2d[i][j];

}

}

findLongestSequence(arr1d, rows \* cols);

for (int i = 0; i < rows; i++) {

free(arr2d[i]);

}

free(arr2d);

free(arr1d);

return 0;