**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**

**Кафедра системного проектування**

**Звіт**

**про виконання лабораторної роботи №8**

**з дисципліни «Алгоритмізація та програмування»**

Виконав:  
студент I курсу, групи ДА-02

Рудік Андрій Миколайович  
Прийняв:

к.т.н., доцент Безносик О. Ю.

Київ – 2020

Завдання

Дана дійсна квадратна матриця n-го порядку. Побудувати послідовність дійсних чисел а1,...,an за правилом: якщо в i-ому рядку матриці елемент, що належить головній діагоналі, від`ємний, то аі дорівнює сумі елементів, які попередні першому від`ємному елементу i-го рядка, інакше аі дорівнює сумі останніх елементів i-го рядка, починаючи з першого по порядку від`ємного елемента.

Аналіз умови задачі

Одна вхідна змінна N — розмірність квадратної матриці. Завдання у достатній мірі описує майбутній алгоритм, а тому не бачу сенсу щось додавати. Блок-схема алгоритму представлена на рисунку 1.

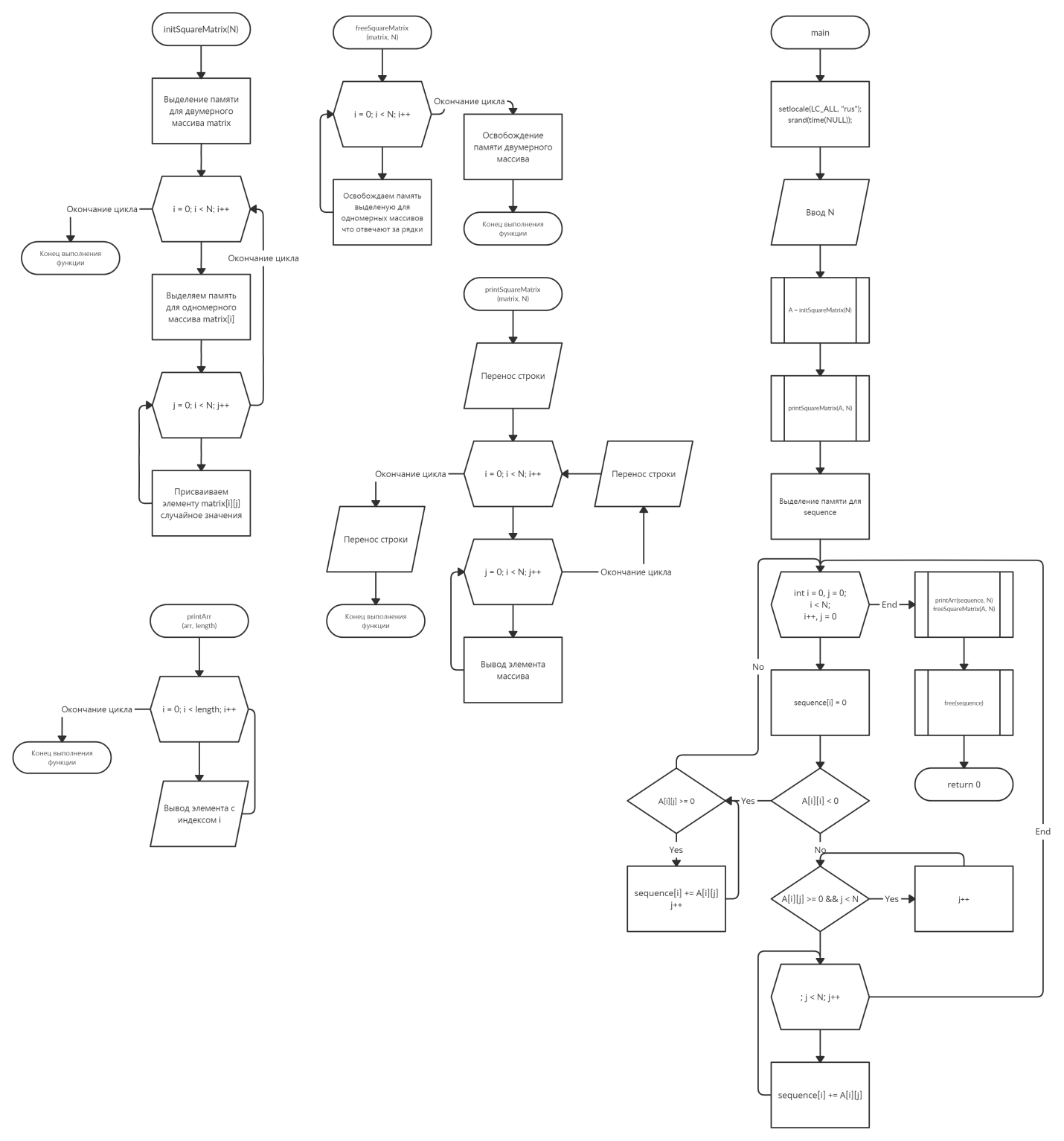


Рисунок 1 – Блок-схема алгоритму

Код програми

#include <locale.h>

#include <stdio.h>

#include <malloc.h>

#include <time.h>

#include <stdlib.h>

#include <math.h>

*void* printArr(*int* \**arr*, *int* *length*) {

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

    printf("%d ", *arr*[i]);

  }

  printf("\n");

}

*int* \*\*initSquareMatrix(*int* *N*) {

*int* \*\*matrix = (*int*\*\*)malloc(*N* \* sizeof(*int*));

  for (*int* i = 0; i < *N*; i++) {

    matrix[i] = (*int*\*)malloc(*N* \* sizeof(*int*));

    for (*int* j = 0; j < *N*; j++) {

      matrix[i][j] = (rand() % 199) - 99; // [-99 ; 99]

    }

  }

  return matrix;

}

*void* printSquareMatrix(*int* \*\**matrix*, *int* *N*) {

  printf("\n");

  for (*int* i = 0; i < *N*; i++) {

    for (*int* j = 0; j < *N*; j++) {

      printf("%3d\t", *matrix*[i][j]);

    }

    printf("\n");

  }

  printf("\n");

}

*void* freeSquareMatrix(*int* \*\**matrix*, *int* *N*) {

  for (*int* i = 0; i < *N*; i++) {

    free(*matrix*[i]);

  }

  free(*matrix*);

}

*int* main() {

  setlocale(LC\_ALL, "rus");

  srand(time(NULL));

*int* N;

  printf("Введите число N, которое отвечает за количество столбцов и строк в матрице: ");

  scanf("%d", &N);

*int* \*\*A = initSquareMatrix(N);

  printSquareMatrix(A, N);

*int* \*sequence = (*int*\*)malloc(N \* sizeof(*int*));

  for (*int* i = 0, j = 0; i < N; i++, j = 0) {

    sequence[i] = 0;

    if (A[i][i] < 0) {

      while (A[i][j] >= 0) {

        sequence[i] += A[i][j];

        j++;

      }

      continue;

    }

    while(A[i][j] >= 0 && j < N) j++;

    for(; j < N; j++) {

      sequence[i] += A[i][j];

    }

  }

  printf("Последовательность созданная на базе матрицы: \n");

  printArr(sequence, N);

  freeSquareMatrix(A, N);

  free(sequence);

  return 0;

}

Результат роботи програми наведено на рисунку 2

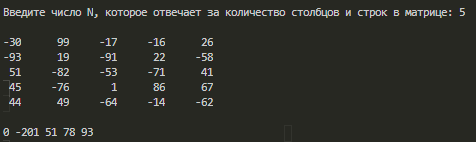


Рисунок 2 – Результат роботи програми

Висновки

В ході виконання лабораторної роботи №8 я закріпив навички роботи з найпростішими алгоритмами.