

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет информатики и  
радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт  
по лабораторной работе № 5  
«Указатель на указатель для работы с многомерными массивами. Указатели.  
Динамическое распределение памяти.»  
по курсу:  
«Основы алгоритмизации и программирования»

Выполнил:  
Студент группы 950503  
Полховский А.Ф.

Проверил:  
Дубовский А.Л.

Минск 2019

Цель работы: Изучить операции работы с указателями и научиться использовать функции динамического выделения памяти для одномерных массивов. Научиться использовать указатель на указатель при работе с двумерными массивами.

#### Вариант 4

а) Рассортировать положительные элементы каждой строки матрицы по убыванию. Отрицательные элементы оставить на своих местах.

б) Дана действительная квадратная матрица порядка  $n$ . Рассмотрим те элементы, которые расположены в строках, начинающихся с отрицательного элемента. Найти суммы этих элементов, которые расположены соответственно ниже, выше и на главной диагонали. Суммы найденных элементов хранить в массиве. Память под массивы выделять динамически.

Код программ на языке Си:

```
а)
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <malloc.h>
#include <locale.h>

//void hoar(int *, int, int); // работает не для всех случаев:(((
//void hoar(int *ms, int left, int right)
//{
//    int k = 0, m = 0, l = 0, i = 0;
//    int srtn;
//    while ((srtn = *(ms + ((left + i + right) / 2))) < 0)
//    {
//        i++;
//        if (i >= right) return;
//    }
//    k = left + i;
//    m = right;
//    while (k <= m)
//    {
//        while (*(ms + k) > srtn || (*(ms + k) < 0)) k++;
//        while (*(ms + m) < srtn || (*(ms + m) < 0)) m--;
//        if (k <= m)
//        {
//            l = *(ms + k);
//            *(ms + k++) = *(ms + m);
//            *(ms + m--) = l;
//        }
//    }
//    if (left < m)
//        hoar(ms, left, m);
//    if (k < right)
//        hoar(ms, k, right);
//}
void hoar(int *, int);
void hoar(int *ms, int n)
{
    int j, i, l = 0, s;
    for (i = 0; i < n; i++)
    {
        if (*(ms + i) < 0) continue;
        for (j = n - 1; i <= j; j--)
```

```

        {
            s = 1;
            while (*(ms + j - s) < 0) s++;
            if (j - s < 0) continue;
            if (*(ms + j - s) <= *(ms + j))
            {
                l = *(ms + j);
                *(ms + j) = *(ms + j - s);
                *(ms + j - s) = l;
            }
        }
    }
}

int main()
{
    setlocale(LC_ALL, "Russian");
    int *ms, i = 0, j = 0, columns, lines;

    printf("Введите количество строк\n");
    do { rewind(stdin); } while (scanf_s("%d", &lines) != 1);

    printf("Введите количество столбцов\n");
    while (scanf_s("%d", &columns) != 1) { rewind(stdin); };

    ms = (int)malloc(lines * columns * sizeof(int));
    for (i = 0; i < lines; i++)
        for (j = 0; j < columns; j++)
        {
            printf("ms[%d][%d] = ", i, j);
            if (scanf_s("%d", (ms + i*columns + j)) != 1) { rewind(stdin); j--; }
        }

    puts("Исходная матрица:");
    for (i = 0; i < lines; i++)
    {
        puts("\n");
        for (j = 0; j < columns; j++)
            printf("%5d", *(ms + i*columns + j));
    }
    for (j = 0; j < lines; j++)
    {
        hoar((ms + j*columns), columns);
    }
    puts("\nПреобразованная матрица:");

    for (i = 0; i < lines; i++)
    {
        puts("\n");
        for (j = 0; j < columns; j++)
            printf("%5d", *(ms + i*columns + j));
    }
    puts("\n");
    system("PAUSE");
    return 0;
}

```

б)

```

#include <malloc.h>
#include <stdio.h>
#include <conio.h>

```

```

int main()
{

```

```

int **ms, **sum, n, i = 0, j = 0;
puts("Please enter n == ");
do { rewind(stdin); } while (scanf_s("%d", &n) != 1);
ms = (int **)malloc(n * sizeof(int*));
sum = (int **)malloc(n * sizeof(int*)); // n - количество строк
for (int j = 0; j < n; j++)
{
    ms[j] = (int*)malloc(n * sizeof(int));
    sum[j] = (int*)calloc(3, sizeof(int)); // 3 - количество столбцов
}
if ((ms == NULL) || (sum == NULL)) return;
//-----Ввод матрицы-----//
printf("%d x %d matrix \n", n, n);
puts("Entering matrix...");
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
        printf("ms[%d][%d] = ", i, j);
        while (scanf_s("%d", &ms[i][j]) != 1) { rewind(stdin); }
    }
//-----Вывод матрицы-----//
puts("\nYour matrix:");
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
        printf("%5d", ms[i][j]);
    printf("\n");
}
//-----Рабочий цикл-----//
for (i = 0; i < n; i++)
{
    if ((ms[i][0]) < 0)
        for (j = 0; j < n; j++)
        {
            if (i == j)
                sum[i][0] = ms[i][j]; //0 строка sum - диагональ
            if (j < i)
                sum[i][2] += ms[i][j]; //2 строка sum - сумма под диагональю
            if (j > i)
                sum[i][1] += ms[i][j]; //1 строка sum - сумма над диагональю
        }
}
for (i = 0; i < n; i++) // цикл по строкам
    free(ms[i]); //освобождение памяти под строку
free(ms);
//-----Распаковочка(Вывод сумм)-----//
puts("Diagonal      Over diagonal      Under diagonal");
puts("\n");
int sum1 = 0, sum2 = 0, sum3 = 0;
for (j = 0; j < n; j++)
{
    printf("for %d-st line: ", j);
    sum1 += sum[j][0];
    printf_s("%12d", sum[j][0]);
    sum2 += sum[j][1];
    printf_s("%12d", sum[j][1]);
    sum3 += sum[j][2];
    printf_s("%12d", sum[j][2]);
    puts("\n");
}
printf("\nAll elements:      On diagonal = [%d]      Under diagonal = [%d]      Over
diagonal = [%d]\n", sum1, sum3, sum2);
for (i = 0; i < n; i++) // цикл по строкам
    free(sum[i]);
free(sum);
system("PAUSE");

```

Вывод: были изучены краткие теоретические сведения по теме “Указатели. Динамическое распределение памяти. Указатель на указатель для работы с многомерными массивами.” (понятия указателя и указателя на указатели и методы работы с ними; функции выделения и перераспределения памяти); были написаны коды программ 4-ого варианта, а также налажены и выполнены.