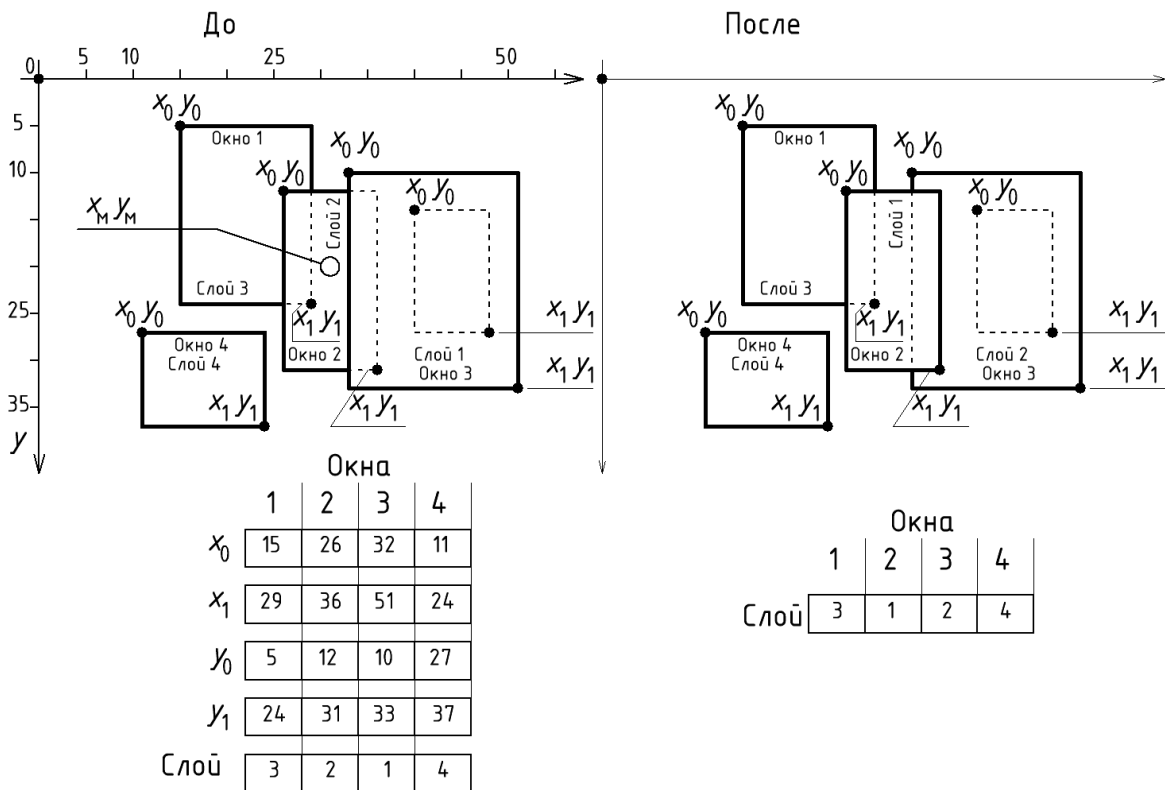


Задание для учащихся 10 -го класса

Окна

Имеется четыре массива содержащих координаты прямоугольников

$A_1 = (x_0, x_1, y_0, y_1), A_2 = (x_0, x_1, y_0, y_1)$ и т. д. Имеется массив слоёв, отражающий расположение окон относительно экрана. Слой 1 соответствует ближнему, последний слой – дальнему. Интерпретация и расположение элементов массива в памяти представлены на рисунке.



Программа получает на вход четыре массива одинаковой длины, состоящих из координат начал и концов отрезков. Элементы массива – целые неотрицательные числа. Координаты $x_1 > x_0, y_1 > y_0$ для любого прямоугольника. Программа получает на вход координаты щелчка мышью x_m, y_m .

Программа должна вывести номера окон в соответствующих слоях. Если мышшь попала по окну – окно становится на первый слой. Порядок остальных слоёв – сохраняется. Вывести номер окна по которому попала мышшь, если мышшь не попала ни в одно окно – вывести сообщение «Координаты вне окон». При правильном использовании структур данных для координат окна – дополнительно 3 балла за решение задачи с использованием структур.

Примечание: Индексация окон начинаются с единицы.

Требования к оформлению заданий

Использовать определения массивов, операторы ввода-вывода, условные операторы, циклы, индексацию элементов массива.

Решение

KT1

```
//координаты окон
int Ax0[] = {15, 26, 32, 11};
int Ax1[] = {29, 36, 51, 24};
int Ay0[] = {5, 12, 10, 27};
int Ay1[] = {24, 31, 33, 37};
int Sloi[] = {3,2,1,4};
int Mx = 31, My = 20;//координаты щелчка мышью
```

При использовании структур данных вводится структура окна:

```
typedef struct Okno
{
int x0;int y0;int x1; int y1;
} Okno;

Okno Okna = {...}, {...}, {...}, {...}; //внутри скобок-заданные
численные параметры
```

KT2

```
//размер массива
int KolichOkon = sizeof(Sloi)/sizeof(int);
```

```

printf ("Расположение окон:\n");
//цикл вывода на экран
for (int i = 0; i < KolichOkon; i++)
{
//аккуратно выводится запятая между элементами
    if (i == (KolichOkon - 1) )
    {
printf ("%d",Sloi[i]);
    }
else
{
    printf ("%d,",Sloi[i]);
}
}
printf ("\n");

```

КТЗ

```

int NomerOkna = 0; // 0 - значит в окно не попали
for (int i = 0; i < KolichOkon; i++)
{
    int Idx0 = Sloi[i] - 1; //индекс текущего окна
    //координаты текущего окна
    int x0 = Ax0[Idx0], x1 = Ax1[Idx0], y0 = Ay0[Idx0],
y1 = Ay1[Idx0];
    //в какое окно попала координата
    //условия для поиска окна

```

```
}
```

КТ4

```
//условия для поиска окна  
  
//ищем только видимое окно (невидимые окна за окнами, и  
окна на последующих слоях - пропускаются)  
  
if (Mx >= x0 && Mx <= x1 && My >= y0 && My <= y1)  
{  
    NomerOkna = i;  
    break;  
}
```

КТ5

Смещение индексов элементов массива слоёв и вывод текущего на экран.

```
//вывести на экран текущий номер  
printf ("текущее окно %d\n", NomerOkna);  
for (int i = KolichOkon - 1; i >= 0; i--)  
{  
    //копируем идентификаторы окон на последующие слои  
    Sloi[i + 1] = Sloi[i];  
}  
  
Sloi[0] = NomerOkna;
```

Ответ (пример при конкретном входном значении)

Вывод:

До нажатия:

Расположение окон по слоям от 1 до 4:

3,2,1,4

Выбрано окно 3.

После нажатия:

2,3,1,4

Выбрано окно 2

Ответ представлен при входных данных в соответствии с таблицей:

Ввод

	1	2	3	4
x_0	15	26	32	11
x_1	29	36	51	24
y_0	5	12	10	27
y_1	24	31	33	37

Координаты щелчка мышью: $x_M = 31, y_M = 20$.

Критерии оценки

- 1) Формируются массивы значений в виде соответствующих переменных. Правильно объявлены переменные - **20 баллов** (КТ1)
- 2) п. 1) плюс умение определить начальные значения, верно задать указатели и/или итераторы, выведен исходный массив окон по слоям. Подобный цикл реализуется для вывода слоёв далее **40 баллов** (КТ1+КТ2)
- 3) п.2) плюс знания, что такое индексация в массиве, каким образом массив может содержать в качестве элементов индексы в другом массиве, знания по

использованию указателей, умение использования идентификатора в качестве индекса массивов – **60 баллов** (КТ1+КТ2+КТ3)

4) п.3) плюс умение записывать условные выражения с несколькими операндами – **80 баллов** (КТ1+КТ2+КТ3+КТ4)

5) п. 4) плюс : умение работать с элементами массива с использованием индексации в виде зависимости соседних элементов - **100 баллов** (КТ1+КТ2+КТ3+КТ4+КТ5)