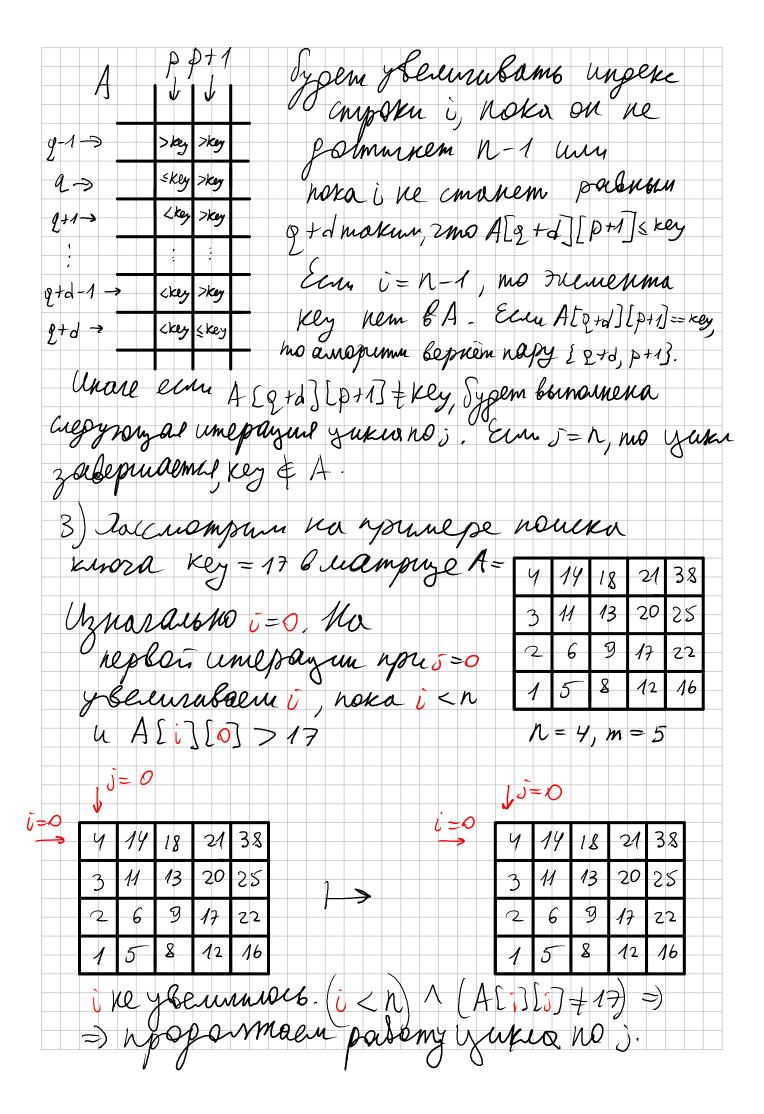
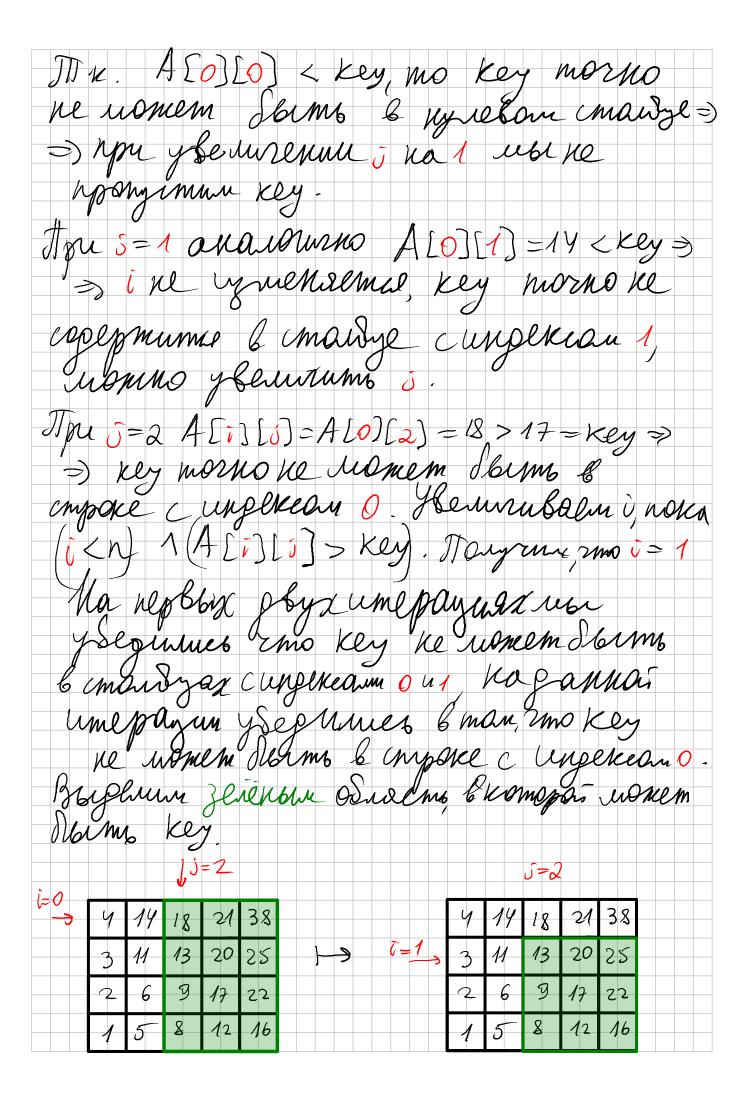
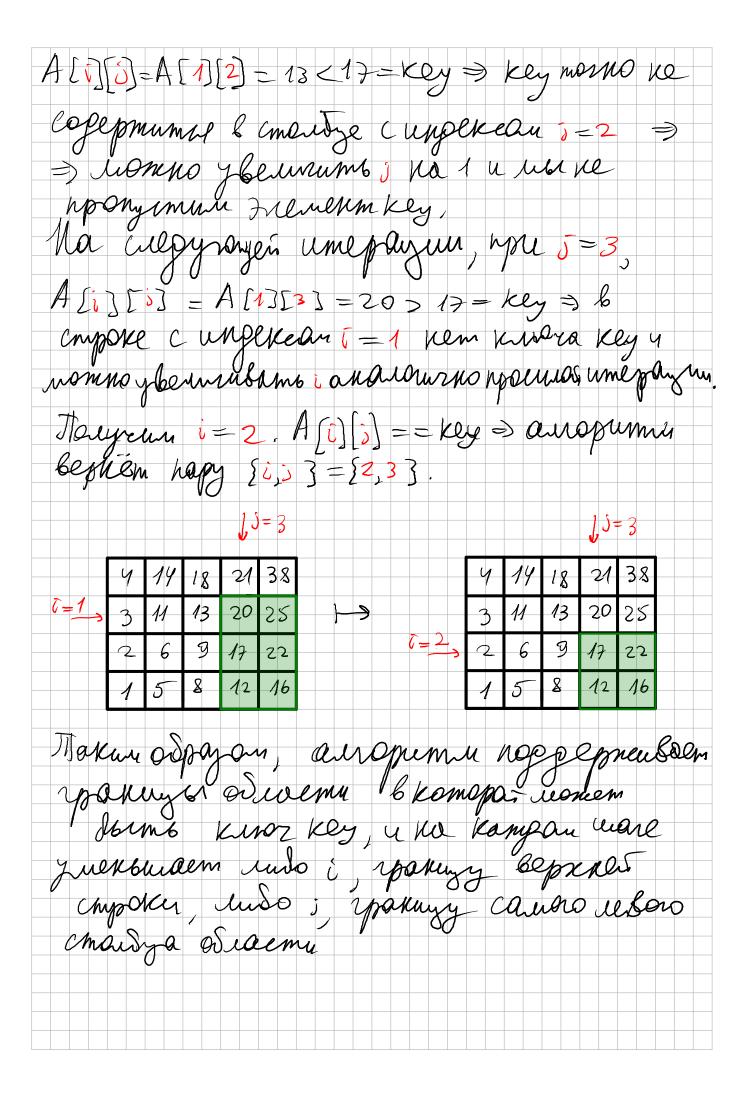
1. a) Fanemun, uno elen A[i][s] > Key, mo bel sienenny A 6 composer i a condedy ax 3+1,5+2,.., n-1 mome Sousue key, m.e. A[i][i]>key =>(tk = i+1, n-1 => A[i][k] > key) Curonembo { 5+1,5+2, -, n-1} Marme, eau ACi][5] > key, mo irodoi dienenm 6 marsje : u compokase i-1, ... o mome barone key me A[i][i]>key => (\frac{1}{k}\in 0, i-1 => A[k][i] > key) 5) Onwen anopumu komopera pemaen 3 agary gree wamping or paymepa n X m (u, 6 raemnamu, gus mampuyon paymepa nxn) Sporgence gukian no jom opom-1 (0 s j < m) ho mones que money en le ka kongois umepaixue dypen noppepmulamo UKPEKE CAYOKU i, MOKOL ZMO IMO replat composed 6 mansure, que les massos A [i][j] < key. Jyong ppi 5 = P makor ungeke i=2, morpa morno yblemus mo A[2][p] < key, a 2=0 chu A[2-1][p] > key. Eun A[2][p] + kes, mo na culgyrought unepayur, upu 5=p+1







Realizague amplumba nel 954 CHT

Полный код этой реализации алгоритма находится конце файла. Работа функции проверялась на версии языка 2020 и 2023 годов с компилятором g++ версии 12.2.0

```
template <typename T>
using matrix_t = std::vector<std::vector<T>>;
/// @brief Searches for the key in the matrix (with special order of the elements).
/// @tparam T type of the element.
/// @param matrix matrix.
/// @param key key to find.
/// @return pair of indexes { row, column } if element is found. Otherwise, returns { (size_t)-1, (size_t)-1 }.
template <typename T>
requires requires (const T& a, const T& b) {
    { a < b } → std::same_as<bool>;
    { a = b } \rightarrow std::same_as<bool>;
std::pair<size_t, size_t> FindKey(const matrix_t<T>& matrix, const T& key) {
    std::pair<size_t, size_t> ans{ static_cast<size_t>(-1), static_cast<size_t>(-1) };
    if (matrix.empty()) {
        return ans;
    size_t n = matrix.size();
    size_t m = matrix[0].size();
    size_t i = 0;
    for (size_t j = 0; j < m; j++) {
    while (i < n && key < matrix[i].at(j)) {</pre>
        if (i = n) {
            return ans;
        if (key = matrix[i][j]) {
            return ans;
    return ans;
```

2. Dokomily, mo fulgarnoù zapary koja A-kbappannas nampena, T(n) = O(n)Ykamen 6 madunge, kakue berenemmhienee zampambe nymner na kampañ empoke 46 merepun kakaro kourremba umepayen ohn benoineromar.

```
Jampanner
                                                                               unexastin
           std::pair<size_t, size_t> ans{ static_cast<si;</pre>
           if (matrix.empty()) {
                                                                           c_2
                                                                                1
                                                                           c_3
                return ans;
                                                                                1
           size_t n = matrix.size();
                                                                           c_4
                                                                               1
           size_t m = matrix[0].size();
                                                                           c_5
                                                                                1
           size_t i = 0:
                                                                           c_6
          for (size_t j = 0; j < m; j++) {
                                                                           c_7
                                                                                m
                while (i < n \&\& key < matrix[i].at(j)) {
                                                                                m+n
                                                                           c_8
                                                                                n-1
                     i++:
                                                                           c_{\mathsf{q}}
                                                                                m-1
                if (i = n) {
                                                                           c_{10}
                                                                           c_{11} \mid m-1
                    return ans;
                if (key = matrix[i][j]) {
                                                                                 m-1
                                                                           c_{12}
                                                                           c_{13} | m-1
                     ans.first = i;
                                                                                 m-1
                     ans.second = j;
                                                                           c_{14}
                     return ans;
                                                                                 m-1
                                                                           c_{15}
           return ans;
                                                                           c_{16} + 1
De Meno Bhympehnero y cikea white no i
Bernoumina ke souce n-1 pag, m x 0 ≤ i< n
 a ogno in oderaments yerobeis bornamenis
mera moro zinkia - i < n. Trobepti b
Yukie While no i bornamenal ne Seree
htm pas, mx xoma de oppu hpolepka
bennemie na Kampon umepayuu
bkennero yukua no ja hpu kampas
yoruman polepke (== true) i y kennenkalmas
kal.
```

```
T(n) = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 m + c_8 (m+n) + c_9 (n-1) + c_{10} (m-1) + c_{11} (m-1) + c_{12} (m-1) + c_{13} (m-1) + c_{14} (m-1) + c_{15} (m-1) + c_{16} = C_1 + C_2 m + C_3 n, where
C_1 = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 - c_9 - c_{10} - c_{11} - c_{12} - c_{13} - c_{14} - c_{15} + c_{16}
C_2 = c_7 + c_8 + c_{10} + c_{11} + c_{12} + c_{13} + c_{14} + c_{15}
C_3 = c_8 + c_9
\Rightarrow T(n) = \mathcal{O}(n+m) = \mathcal{O}(\max(n,m))
\text{To yurden baghed Manying } A - \text{Kbay pannal}
\Rightarrow n = m \implies T(n) = \mathcal{O}(\max(n,n)) = \mathcal{O}(n)
```

```
Полный код реализации алгоритма.
#include <cstdint> // size_t
#include <concepts> // std::same_as<T>
#include <utility> // std::pair<T, U>
                    // std::vector<T, ...>
#include <vector>
template <typename T>
using matrix_t = std::vector<std::vector<T>>>;
/// \emptysetbrief Searches for the key in the matrix (with special order of the elements).
/// @tparam T type of the element.
/// @param matrix matrix.
/// @param key key to find.
/// @return pair of indexes { row, column } if element is found. Otherwise, returns { (size_t)-1, (size_t)-1 }.
template <typename T>
requires requires (const T& a, const T& b) {
    { a < b } \rightarrow std::same_as<bool>;
    { a = b } \rightarrow std::same_as<bool>;
std::pair<size_t, size_t> FindKey(const matrix_t<T>& matrix, const T& key) {
    std::pair<size_t, size_t> ans{ static_cast<size_t>(-1), static_cast<size_t>(-1) };
    if (matrix.empty()) {
        return ans;
    size_t n = matrix.size();
    size_t m = matrix[0].size();
    size_t i = 0;
    for (size_t j = 0; j < m; j++) {
        while (i < n && key < matrix[i].at(j)) {
        }
        if (i = n) {
            return ans;
        if (key = matrix[i][j]) {
            ans.first = i;
            ans.second = j;
            return ans;
        }
    }
    return ans;
}
```