

1 Правила оформления кода

1. Отступы — два пробела, четыре пробела, либо табуляция. Внутри одного файла только один из видов отступов.
2. Логические и синтаксические блоки отделяются пустой строкой.
3. Имена переменных, функций, констант записываются в camelCase: первое слово строчными буквами, все последующие — с заглавной. Подчёркивания в записи не допускаются. Макроопределения записываются заглавными буквами через подчёркивание.
4. Внутри арифметических и логических выражений пробелами с обеих сторон отделяются бинарные операторы. Пример: `a + b < c / 2` и `x == 1`. Унарные операторы не отделяются пробелом от операнда. Примеры: `x := -x`, `x = *a.begin()`. После открывающей и перед закрывающей скобкой пробел не ставится.
5. Двоеточие и запятая отделяется пробелом справа и не отделяется слева.
6. Условие в конструкциях `if`, `while`, `switch` и других отделяется пробелом от соответствующего ключевого слова.
7. Каждая команда записывается в отдельной строке. Блоки команд (`begin..end` либо фигурные скобки) присутствуют всегда, даже если оператор один. `else` записывается в той же строке, где заканчивается блок команд `if`. Например:

```
if (x < a[av]) {
    r = av;
} else {
    l = av;
}
```

8. При объявлении и вызове функций список параметров/аргументов в скобках не отделяется от имени функции.
9. `begin` или открывающая фигурная скобка записывается на той же строке, что и оператор (исключение: `begin` как начало функции).
10. `var` и `const` в началах блоков записываются на отдельной строке.
11. Оператор присваивания отделяется с обеих сторон пробелами.
12. Для Паскаля: после последнего оператора в блоке `begin..end` всегда ставится точка с запятой.
13. Если в программе используются записи (в Паскале) или классы и структуры (в C++), то они должны быть объявлены и именованы.

Пример кода задачи на C++

```
#include <cstdio>

const int maxN = 100000;

struct structName {
    int value;
};

int n, k;
int a[maxN];
structName b[maxN];

int binSearch(int x) {
    int l = 0, r = n, av;
    while (r - l > 1) {
        av = (l + r) / 2;
        if (x < a[av]) {
            r = av;
        } else {
            l = av;
        }
    }

    if (a[l] == x) {
        return l;
    } else {
        return -1;
    }
}

int main() {
    freopen("right.in", "r", stdin);
    freopen("right.out", "w", stdout);

    scanf("%d%d", &n, &k);
    for (int i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }

    for (int i = 0; i < k; i++) {
        int x;
        scanf("%d", &x);
        printf("%d\n", binSearch(x) + 1);
    }

    return 0;
}
```

Пример кода задачи на Pascal

```
const
  maxN = 100000;

type recordName = record
  value: longint;
end;

var
  n, k: longInt;
  a: array[1..maxN] of longInt;
  b: array[1..maxN] of recordName;

function binSearch(x: longInt): longInt;
var
  l, r, av: longInt;
begin
  l := 1;
  r := n + 1;
  while r - l > 1 do begin
    av := (l + r) div 2;
    if x < a[av] then begin
      r := av;
    end else begin
      l := av;
    end;
  end;
  if a[l] = x then begin
    binSearch := l;
  end else begin
    binSearch := 0;
  end
end;

var
  i, x: longInt;

begin
  assign(input, 'right.in');
  reset(input);
  assign(output, 'right.out');
  rewrite(output);
  readLn(n, k);
  for i := 1 to n do begin
    read(a[i]);
  end;
end;
```

```
for i := 1 to k do begin
  read(x);
  writeln(binSearch(x));
end;
end.
```