```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main(void) {
  setlocale(LC_ALL, "RU");

  int array[10];

  return EXIT_SUCCESS;
}
```

---

```c
#define ARRAY_SIZE 10
```

---

```c
  array[0] = 1;
  ...
  printf("%d\n", array[5]);
  ...
  ++array[i];
```

---

```c
  array[i + j * 10] = 0;

  i = 0;
  while (i < size) {
    array[i++] = 0;
  }
```

## Базовые конструкции

Обнуление массива

```
for (i = 0; i < size; i++) {
  array[i] = 0;
}
```

Заполнение массива

```
for (i = 0; i < size; i++) {
  scanf("%d", &array[i]);
}
```

Суммирование элементов массива

```
for (i = 0; i < size; i++) {
  sum += array[i];
}
```

```c
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 5

int main(void) {
  int array[ARRAY_SIZE];

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

  puts("");

  for (int i = ARRAY_SIZE - 1; i >= 0; i--) {
    printf("%d\n", array[i]);
  }

  return EXIT_SUCCESS;
}
```

---

```c
  int array[ARRAY_SIZE] = {1, 2, 3, 4, 5};

  int array[ARRAY_SIZE] = {1, 2, 3};

  int array[ARRAY_SIZE] = {0};

  int array[] = {1, 2, 3, 4, 5};

  int array[ARRAY_SIZE] = {[1] = 5, [3] = 5};

  int array[ARRAY_SIZE] = {[4] = 5, [2] = 5};

  int array[] = {[7] = 5, [2] = 5};

  int array[ARRAY_SIZE] = {1, [4] = 5, [2] = 5};
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <stdbool.h>

#define ARRAY_SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  bool digitSeen[ARRAY_SIZE] = {false};
  int digit;
  long number;

  puts("Введите число:");
  scanf("%ld", &number);

  while (number > 0) {
    digit = number % 10;
    if (digitSeen[digit]) {
      break;
    }
    digitSeen[digit] = true;
    number /= 10;
  }

  if (number > 0) {
    puts("\nЕсть повторяющиеся цифры.");
  }
  else {
    puts("\nНет повторяющихся цифр.");
  }

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main(void) {
  setlocale(LC_ALL, "RU");

  int array[] = {[7] = 5, [2] = 5};

  int numberOfElements = sizeof(array) / sizeof(array[0]);

  for (int i = 0; i < numberOfElements; i++) {
    printf("%d\n", array[i]);
  }

  return EXIT_SUCCESS;
}
```

---

```c
#define ARRAY_SIZE sizeof(array) / sizeof(array[0])
```

---

```c
  int *end = array + numberOfElements;

  printf("array: %llX\n", (unsigned long long)array);

  for (int *pointer = array; pointer != end; pointer++) {
    printf("%d\n", *pointer);
  }
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main(void) {
  setlocale(LC_ALL, "RU");

  int matrix[5][7];

  return EXIT_SUCCESS;
}
```

---

```c
#define SIZE 5

  int identityMatrix[SIZE][SIZE];

  for (int row = 0; row < SIZE; row++) {
    for (int column = 0; column < SIZE; column++) {
      if (row == column) {
        identityMatrix[row][column] = 1;
      }
      else {
        identityMatrix[row][column] = 0;
      }
    }
  }

  for (int row = 0; row < SIZE; row++) {
    for (int column = 0; column < SIZE; column++) {
      printf("%d ", identityMatrix[row][column]);
    }
    puts("");
  }
```

```
int identityMatrix[SIZE][SIZE] = {{1, 0, 1, 0, 1},
                                  {0, 2, 1},
                                  [2][2] = 7};

                          ---

const char hexChars[] = {'0', '1', '2', '3', '4', '5',
                         '6', '7', '8', '9', 'A', 'B',
                         'C', 'D', 'E', 'F'};
```