

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

#define SIZE 5

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

    int matrix[SIZE][SIZE] = { 0 };
    int upperBound = 100;
    int lowerBound = -upperBound;

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[i][j] = lowerBound + rand() % (upperBound - lowerBound + 1);
        }
    }

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            printf("%-5d ", matrix[i][j]);
        }
        puts("");
    }

    return EXIT_SUCCESS;
}
```

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

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

    char firstString[] = "Привет";

    puts(firstString);

    firstString[5] = '!';

    puts(firstString);

    char secondString[] = {'П', 'р', 'и', 'в', 'е', 'т', '\0'};

    char thirdString[20];

    scanf("%19s", thirdString);

    return EXIT_SUCCESS;
}
```

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

#define SIZE 20

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

    char firstString[SIZE];
    char secondString[] = "строковый литерал";

    printf("%s", "Введите строку (не более 19 символов): ");
    scanf("%19s", firstString);

    printf("\nfirstString: %s\nsecondString: %s\n"
           "firstString с пробелами между символами:\n",
           firstString, secondString);

    for (size_t i = 0; i < SIZE && firstString[i] != '\0'; ++i) {
        printf("%c ", firstString[i]);
    }

    puts("\n");

    return EXIT_SUCCESS;
}
```

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

double CalculateAverage(int a, int b) {
    return (a + b) / 2.;
}

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

    int x = 7, y = 2;

    double average = CalculateAverage(x, y);

    printf("Среднее арифметическое: %g\n", average);

    printf("Среднее арифметическое: %g\n", CalculateAverage(x, y));

    printf("Среднее арифметическое: %g\n", CalculateAverage(7, 2));

    printf("Среднее арифметическое: %g\n", CalculateAverage(y * 2 + 3, -1 + x / 2));

    return EXIT_SUCCESS;
}
```

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

void ShowNumber(int number) {
    printf("%d\n", number);
}

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

    for (int i = 10; i > 0; i--) {
        ShowNumber(i);
    }

    return EXIT_SUCCESS;
}

---
```

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

void ShowMenu(void) {
    puts("Меню");
}

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

    ShowMenu();

    return EXIT_SUCCESS;
}
```

## Общие сведения о функциях

```
return-type function-name ( parameters ) {  
  
    declarations  
  
    statements  
  
}
```

## Общие сведения о функциях

Функции должны иметь определение и объявление, хотя определение может служить объявлением, если находится до вызова функции. Определение функции включает тело функции – код, который выполняется при вызове функции.

Объявление функции указывает имя, возвращаемый тип и атрибуты функции, определенной в любом другом месте программы. Объявление функции должно предшествовать вызову функции.

В вызове функции управление выполнением передается из вызывающей функции в вызываемую функцию. При выполнении оператора **return** в вызываемой функции управление и, возможно, значение возвращаются в вызывающую функцию.

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

```
bool IsPrime(int number) {
    if (number <= 1) {
        return false;
    }

    for (int divisor = 2; divisor * divisor <= number; divisor++) {
        if (number % divisor == 0) {
            return false;
        }
    }

    return true;
}
```

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

    int number = 0;
    scanf("%d", &number);

    if (IsPrime(number)) {
        puts("Число является простым");
    }
    else {
        puts("Число не является простым");
    }

    return EXIT_SUCCESS;
}
```



```

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

double CalculateAverage(double a, double b);

double CalculateAverage(double, double);

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

    int x = 7, y = 2;

    double average = CalculateAverage(x, y);

    printf("Среднее арифметическое: %g\n", average);

    printf("Среднее арифметическое: %g\n", CalculateAverage(x, y));

    printf("Среднее арифметическое: %g\n", CalculateAverage(7, 2));

    printf("Среднее арифметическое: %g\n", CalculateAverage(y * 2 + 3, -1 + x / 2));

    return EXIT_SUCCESS;
}

double CalculateAverage(double a, double b) {
    return (a + b) / 2;
}

```

## Объявление функции

*return-type function-name ( parameters ) ;*

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

int SumArrayElements(int array[], int size) {
    int sum = 0;

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

    return sum;
}

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

    int array[] = {[7] = 4, [2] = 5};

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

    int total = SumArrayElements(array, numberOfElements);

    printf("Сумма элементов: %d\n", total);

    return EXIT_SUCCESS;
}
```

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

int SumArrayElements(int array[], int size) {
    int sum = 0;

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

    return sum;
}

void ZeroingArray(int array[], int size) {
    for (int i = 0; i < size; i++) {
        array[i] = 0;
    }
}

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

    int array[] = {[7] = 4, [2] = 5};

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

    int total = SumArrayElements(array, numberOfElements);

    printf("Сумма элементов до: %d\n", total);

    ZeroingArray(array, numberOfElements);

    total = SumArrayElements(array, numberOfElements);

    printf("Сумма элементов после: %d\n", total);

    return EXIT_SUCCESS;
}
```

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

#define ROWS 3
#define COLUMNS 3

int SumArrayElements(int array[][COLUMNS]) {
    int sum = 0;

    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            sum += array[i][j];
        }
    }

    return sum;
}

void ZeroingArray(int array[][COLUMNS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            array[i][j] = 0;
        }
    }
}

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

    int matrix[][COLUMNS] = {{1, 0, 1},
                                {0, 2, 1},
                                [2][2] = 7};

    int total = SumArrayElements(matrix);

    printf("Сумма элементов до: %d\n", total);

    ZeroingArray(matrix);

    total = SumArrayElements(matrix);

    printf("Сумма элементов после: %d\n", total);

    return EXIT_SUCCESS;
}
```

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

int SumArrayElements(int array[], int size) {
    int sum = 0;

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

    return sum;
}

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

    int size = 8;

    int total = SumArrayElements((int []){[7] = 4, [2] = 5}, size);

    printf("Сумма элементов: %d\n", total);

    return EXIT_SUCCESS;
}
```

```

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

unsigned long long GetFibonacciNumberWithRecursion(unsigned int);

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

    unsigned int number;
    scanf("%u", &number);

    unsigned long long result = GetFibonacciNumberWithRecursion(number);

    printf("Число Фибоначчи № %u = %llu\n", number, result);

    return EXIT_SUCCESS;
}

unsigned long long GetFibonacciNumberWithRecursion(unsigned int number) {
    if (0 == number || 1 == number) {
        return number;
    }
    else {
        return GetFibonacciNumberWithRecursion(number - 1)
            + GetFibonacciNumberWithRecursion(number - 2);
    }
}

```

```

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

unsigned long long GetFibonacciNumberWithIteration(unsigned int);

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

    unsigned int number;
    scanf("%u", &number);

    unsigned long long result = GetFibonacciNumberWithIteration(number);

    printf("Число Фибоначчи № %u = %llu\n", number, result);

    return EXIT_SUCCESS;
}

unsigned long long GetFibonacciNumberWithIteration(unsigned int number) {
    unsigned long long first = 0, second = 1;
    unsigned long long next = 0;

    if (0 == number || 1 == number) {
        return number;
    }

    for (unsigned int i = 2; i <= number; ++i) {
        next = first + second;
        first = second;
        second = next;
    }

    return next;
}

```