

# Программирование

## Лабораторная работа № 2

Карпов Р. В., ИВТ-2

### 1. Тема лабораторной работы

Указатели, арифметика указателей

### Задание 1

### 2. Постановка задачи

Внутри функции `int main(void) { /*...*/ }` определите указатель `double ***pointer = NULL;`. Инициализируйте этот указатель адресом другого указателя типа `double **`, который указывает на переменную `double *`, которая указывает на `double`. Используйте `pointer` для записи и чтения значения 2.0 в сегмент оперативной памяти для `double`.

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
p	double*	Указатель на выделенную память размера double в памяти
p1	double**	Указатель на выделенную память размера double* в памяти
pointer	double***	Указатель на выделенную память размера double** в памяти

## 5. Код программы

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

int main() {
    double* p = malloc(sizeof(double));
    double** p1 = malloc(sizeof(double*));
    double*** pointer = malloc(sizeof(double**));

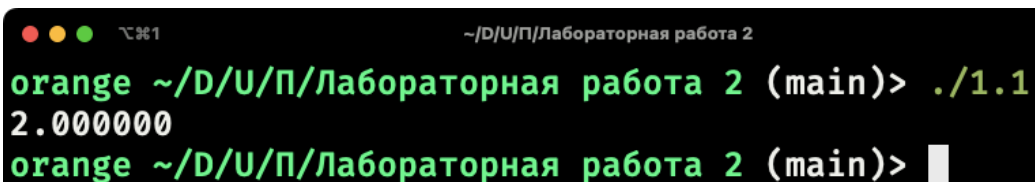
    *p1 = p;
    *pointer = p1;
    ***pointer = 2.0;

    printf("%Lf\n", *p);

    free(p);
    free(p1);
    free(pointer);

    return 0;
}
```

## 6. Результаты выполненной работы

A terminal window with a black background and green text. The title bar shows three colored circles (red, yellow, green) and the text "orange". The terminal path is "~/D/U/П/Лабораторная работа 2". The prompt is "orange ~/D/U/П/Лабораторная работа 2 (main)>". The user enters "./1.1" and the output is "2.000000". The prompt is then "orange ~/D/U/П/Лабораторная работа 2 (main)>".

```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.1
2.000000
orange ~/D/U/П/Лабораторная работа 2 (main)> 
```

## Задание 2

### 2. Постановка задачи

Напишите программу, которая складывает два числа с использованием указателей на эти числа.

### 3. Математическая модель

$$result = a + b$$

### 4. Список идентификаторов

Имя	Тип	Смысл
a	int	Число a
b	int	Число b
result	int	Результат сложения a и b
pa	int*	Указатель на число a
pb	int*	указатель на число b

## 5. Код программы

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 10;

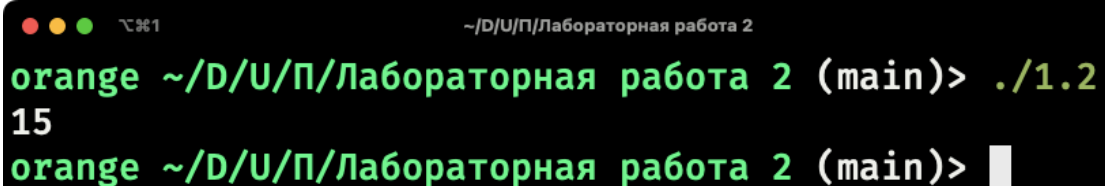
    int* pa = &a;
    int* pb = &b;

    int result = *pa + *pb;

    printf("%d\n", result);

    return 0;
}
```

## 6. Результаты выполненной работы



```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.2
15
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```

## Задание 3

### 2. Постановка задачи

Напишите программу, которая находит максимальное число из двух чисел, используя указатели на эти числа.

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
a	int	Число a
b	int	Число b
pa	int*	Указатель на число a
pb	int*	Указатель на число b
max	int*	Максимальное число
pmax	int*	Указатель на максимальное число

## 5. Код программы

```
#include <stdio.h>

int main() {
    int a = 50;
    int b = 100;

    int* pa = &a;
    int* pb = &b;

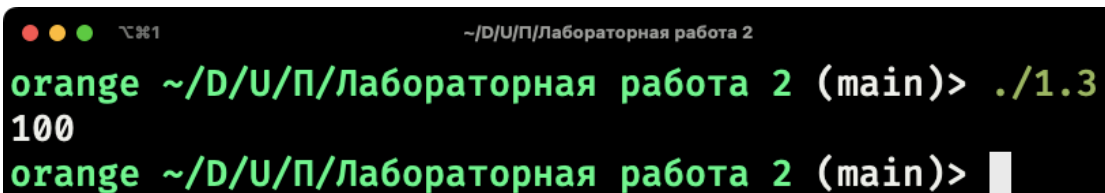
    int max = 0;
    int* pmax = &max;

    if (*pa > *pb) {
        *pmax = *pa;
    } else {
        *pmax = *pb;
    }

    printf("%d\n", *pmax);

    return 0;
}
```

## 6. Результаты выполненной работы



```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.3
100
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```

## Задание 4

### 2. Постановка задачи

Напишите программу, которая создаёт одномерный динамический массив из чисел с плавающей точкой двойной точности, заполняет его значениями с клавиатуры и распечатывает все элементы этого массива, используя арифметику указателей (оператор +), а не оператор доступа к элементу массива [].

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
size	size_t	Размер массива
array	double*	Указатель на выделенную память под массив

## 5. Код программы

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

int main() {
    size_t size;

    printf("Vvedite kolichestvo elementov v massive: ");
    scanf("%zu", &size);

    double* array = (double*) malloc(size * sizeof(double));

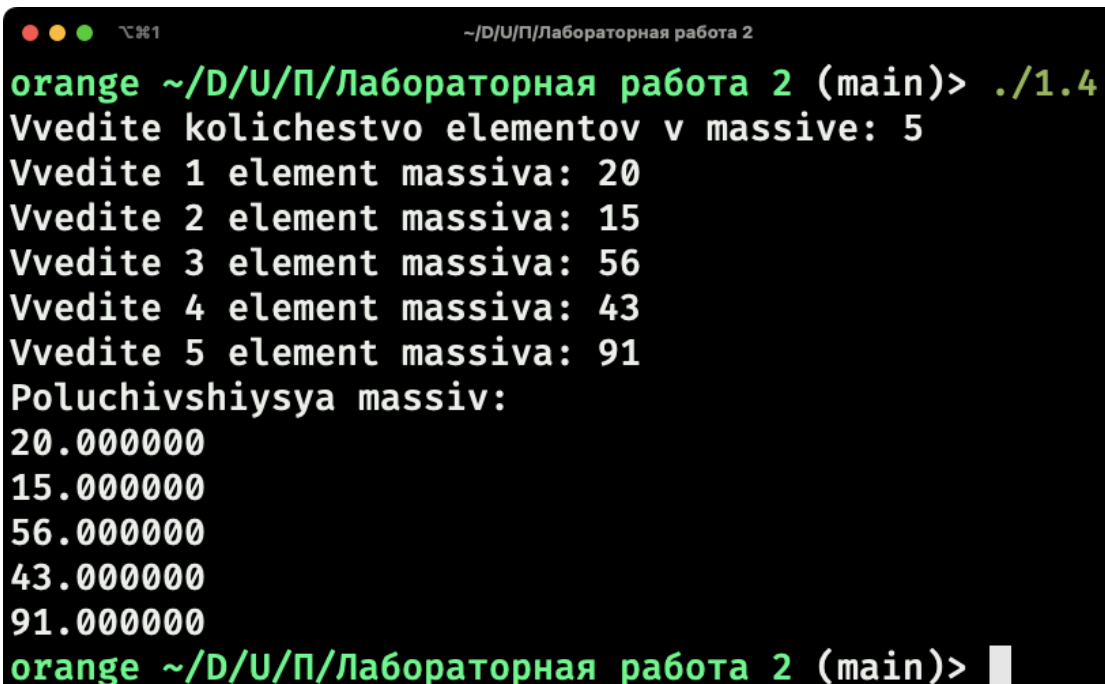
    for (size_t i = 0; i < size; ++i) {
        printf("Vvedite %zu element massiva: ", i + 1);
        scanf("%lf", array + i * sizeof(double));
    }

    printf("Poluchivshiysya massiv:\n");
    for (size_t i = 0; i < size; ++i) {
        printf("%lf\n", *(array + i * sizeof(double)));
    }

    free(array);

    return 0;
}
```

## 6. Результаты выполненной работы



```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.4
Vvedite kolichestvo elementov v massive: 5
Vvedite 1 element massiva: 20
Vvedite 2 element massiva: 15
Vvedite 3 element massiva: 56
Vvedite 4 element massiva: 43
Vvedite 5 element massiva: 91
Poluchivshiysya massiv:
20.000000
15.000000
56.000000
43.000000
91.000000
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```



## Задание 5

### 2. Постановка задачи

Выведите элементы динамического массива целых чисел в обратном порядке, используя указатель и операцию декремента (--).

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
size	size_t	Размер выделяемого массива
array	double*	Указатель на выделенную память под массив
parray	double*	Второй указатель на выделенную память под массив для которого будет применяться арифметика указателей.

## 5. Код программы

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

int main() {
    size_t size;

    printf("Vvedite kolichество elementov v massive: ");
    scanf("%zu", &size);

    double* array = (double*) malloc(size * sizeof(double));
    double* parray = array;

    for (size_t i = 0; i < size; ++i) {
        printf("Vvedite %zu element massiva: ", i + 1);
        scanf("%lf", parray);
        ++parray;
    }

    printf("Poluchivshiysya massiv v obratnom poryadke:\n");
    for (size_t i = 0; i < size; ++i) {
        --parray;
        printf("%lf\n", *parray);
    }

    free(array);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.5
Vvedite kolichество elementov v massive: 5
Vvedite 1 element massiva: 43
Vvedite 2 element massiva: 52
Vvedite 3 element massiva: 86
Vvedite 4 element massiva: 23
Vvedite 5 element massiva: 10
Poluchivshiysya massiv v obratnom poryadke:
10.000000
23.000000
86.000000
52.000000
43.000000
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```

## Задание 6

### 2. Постановка задачи

Определите переменную целого типа `int a = 1234567890`; и выведите побайтово её содержимое на экран, используя указатель `char*`.

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
a	int	Число
pa	char*	Указатель типа char на число a

## 5. Код программы

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

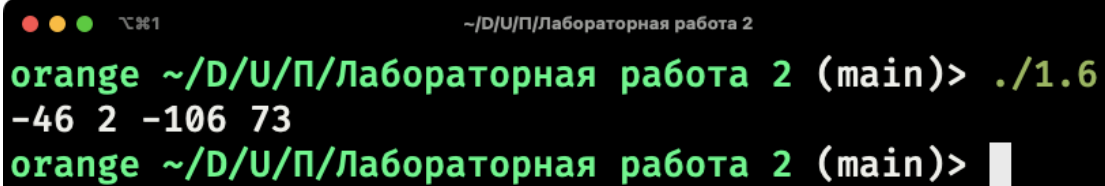
int main() {
    int a = 1234567890;

    char* pa = (char*) &a;

    for (size_t i = 0; i < sizeof(a); ++i) {
        printf("%d ", pa[i]);
    }
    printf("\n");

    return 0;
}
```

## 6. Результаты выполненной работы



```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.6
-46 2 -106 73
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```

## Задание 7

### 2. Постановка задачи

Выделите память под двумерный динамический массив, используя массив указателей на строки (см. лекции), и затем корректно освободите оперативную память.

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
rows	size_t	Количество строк
cols	size_t	Количество столбцов
array	double**	Указатель на выделенную память под массив из указателей на double

## 5. Код программы

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

int main() {
    size_t rows, cols;

    printf("Vvedite kolichestvo strok: ");
    scanf("%zu", &rows);

    printf("Vvedite kolichestvo stolbtsov: ");
    scanf("%zu", &cols);

    double **array = malloc(rows * sizeof(double*));
    for (size_t row = 0; row < rows; ++row) {
        array[row] = (double*) malloc(cols * sizeof(double));
    }

    for (size_t row = 0; row < rows; ++row) {
        for (size_t col = 0; col < cols; ++col) {
            printf("Vvedite element dlya %zu stroki, %zu stolbtsa: ", row
+ 1, col + 1);
            scanf("%lf", &array[row][col]);
        }
    }

    printf("Matritsa:\n");
    for (size_t row = 0; row < rows; ++row) {
        for (size_t col = 0; col < cols; ++col) {
            printf("%lf ", array[row][col]);
        }
        printf("\n");
    }

    for (size_t row = 0; row < rows; ++row) {
        free(array[row]);
    }

    free(array);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 2 (main)> ./1.7
Vvedite kolichество strok: 3
Vvedite kolichество stolbtsov: 3
Vvedite element dlya 1 stroki, 1 stolbtsa: 1
Vvedite element dlya 1 stroki, 2 stolbtsa: 2
Vvedite element dlya 1 stroki, 3 stolbtsa: 3
Vvedite element dlya 2 stroki, 1 stolbtsa: 4
Vvedite element dlya 2 stroki, 2 stolbtsa: 5
Vvedite element dlya 2 stroki, 3 stolbtsa: 6
Vvedite element dlya 3 stroki, 1 stolbtsa: 7
Vvedite element dlya 3 stroki, 2 stolbtsa: 8
Vvedite element dlya 3 stroki, 3 stolbtsa: 9
Matritsa:
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
7.000000 8.000000 9.000000
orange ~/D/U/П/Лабораторная работа 2 (main)> █
```