# Лабораторная работа №2-11, 2-12. Работа с адресами. Работа с указателями

*Задание:*

*Задание 1.*

1. Есть ли среди приведенных определений недопустимые?

a) int \*ip;

b) char \*s = 0;

c) int i;

double\* dp = i;

d) int \*ip, \*ip2;

e) float \*pf = Null;

2. Что выполняет следующий код:

int i = 42, j = 1024;

int \*p1 = &i, \*p2 = &j

\*p2 = \*p1 \* \*p2;

\*p1 \*= \*p1;

*Задание 2.*

[1], Модуль 1. Легкие. Урок 1.3. Указатели

1.3.1 https://stepik.org/lesson/191889/step/1?unit=166415

1.3.2 https://stepik.org/lesson/191889/step/2?unit=166415

1.3.3 https://stepik.org/lesson/191889/step/3?unit=166415

Задание 3.

Разработать 3 любых программы (из ЛЗ-02 с использованием указателей)

*Ожидаемый результат (файлы в своей ветке на гитхабе)*

1. Ответы на тесты

Файлы проектов в своей ветке на гитхабе

*Источники*:

1. Tatyana Berlenko. Тренажёр для решения задач на языке С  
   <https://stepik.org/course/44687>
2. Файлы 4-1.c, 3-3.c

*Файлы*

**4-1.c**

/\* Изменение значения переменной при помощи указателя (4-1.c) \*/

#include <stdio.h>

int main()

{

int a1 = 5, a2 = 7; /\* создаем две переменные \*/

int \*pa = &a1; /\* присваиваем указателю pa адрес переменной \*/

\*pa = a2; /\* изменяем значение переменной, на которую указывает указатель \*/

printf("%d", \*pa); /\* вывод: 7 \*/

return 0;

}

**3-3.c**

/\*\* выражения с указателями (3-3.c) \*/

#include <stdio.h>

int main()

{

/\*\* создаем 2 переменных и 2 указателя \*/

int a = 3, b = 0, \*c = &b, \*d;

d = c; /\*\* указателю d присваиваем адрес указателя c \*/

/\*\* в цикле:указателю d присваиваем адрес переменной a

условие \*c < \*d (эквивалент b < a)

(\*d)++ эквивалент a = a + 1

\*с += 2 эквивалент b = b + 2 \*/

for (d = &a; \*c < \*d; (\*d)++, \*c += 2)

printf("a=%d b=%d\n", a, b);

/\*\* вывод:

a=3 b=0

a=4 b=2

a=5 b=4

\*/

return 0;

}