

Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №3**  
по дисциплине «Информатика: Основы программирования»  
на тему «Указатели»

Выполнил:  
Студент Смирнов И.Д.  
Группа О719Б

Преподаватель:  
Назарова М.А.

Санкт-Петербург  
2021 г.

## Задание 1.

Проанализировать текст представленной программы и выдаваемые программой результаты. Объяснить, почему результаты именно такие.

Результаты работы программы:

```
/home/ivan/Practice/practice3/task1
a:   int: start address 0x7ffcf3387c54 extent 4
b:   float: start address 0x7ffcf3387c50 extent 4
c:   double: start address 0x7ffcf3387c48 extent 8

p1: pointer: start address 0x7ffcf3387c40 extent 8
p2: pointer: start address 0x7ffcf3387c38 extent 8
p3: pointer: start address 0x7ffcf3387c30 extent 8

p1: 0x7ffcf3387c54 related value 1
p2: 0x7ffcf3387c50 related value 2.000000
p3: 0x7ffcf3387c48 related value 3.000000

a=1      b=2.000000      c=3.000000
a=5      b=10.000000     c=1.732051
*p1=5    *p2=10.000000  *p3=1.732051

p1=0x7ffcf3387c50      p2=0x7ffcf3387c50      p3=0x7ffcf3387c50      p4=0x7ff
cf3387c50
*p1=1092616192 *p2=10.000000 *p3=0.000000 *(float*)p4=10.000000

p1=0x7ffcf3387c54      p2=0x7ffcf3387c50      p3=0x7ffcf3387c48
*p1=5 *p2=10.000000 *p3=1.732051
p1=0x7ffcf3387c44      p2=0x7ffcf3387c50      p3=0x7ffcf3387c4c
*p1=32764 *p2=10.000000 *p3=524288.124967

Process returned 0 (0x0)   execution time : 0.006 s
Press ENTER to continue.
—
```

Текст программы:

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

```
int main()
```









```

{
    char *p, c;
    int *a, b;
    float *x, y = 3.5;
    double *m, n;
    a = &b;
    printf("    Enter b = ");
    scanf("%d", a);
    printf("p=%p\tc=%p\ta=%p\tb=%p\tx=%p\ty=%p\tm=%p\tn=%p\n\n", &p, &c, &a, &b, &x, &y, &m, &n);
    printf("a=%p\t*a=%d\tb=%d\n", a, *a, b);
    p = (char*)a;
    c = *p;
    *p = *(p+3);
    *(p+3) = c;
    printf("p=%p\tc=%d\ta=%p\tb=%d\n", p, c, a, b);
    x = &y;
    printf("x=%p\t*x=%f\ty=%f\n", x, *x, y);
    a = (int*)x;
    *a = *x;
    printf("a=%p\t*a=%d\tx=%p\t*x=%f\ty=%f\n", a, *a, x, *x, y);
    a = &b;
    y = 12345.6789;
    printf("x=%p\t*x=%f\ty=%f\n", x, *x, y);
    p = (char*)x;
    c=*p;
    *p=*(p+3);
    *(p+3)=c;
    printf("p=%p\tc=%d\tx=%p\ty=%f\n", p, c, x, y);
    m = &n;
    printf("m=%p\t*m=%lf\tn=%lf\n", m, *m, n);
    n = 5.5;
    printf("m=%p\t*m=%lf\tn=%lf\n", m, *m, n);
    b = n = y = 1.7;
    printf("b=%d\ty=%f\tn=%lf\n", b, y, n);
    printf("*a=%d\t*x=%f\t*m=%lf\n", *a, *x, *m);
    m++;
    printf("n=%lf\tn=%p\tm=%p\n", n, &n, m);
    *m = (float)*a - n + (int)*x;
    printf("m=%p\t*m=%lf\n\n", m, *m);
    printf("p=%p\tc=%d\ta=%p\tb=%d\tx=%p\ty=%f\tm=%p\tn=%f\n\n", p, c, a, b, x, y, m, n);
    return 0;
}

```

Результаты работы программ:

первый вариант (m+=2;)

```
Enter b = 1
p=0000000000061FE18    c=0000000000061FE17    a=0000000000061FE08    b=0000000000061FE04    x=0000000000061FDF8
    y=0000000000061FDF4    m=0000000000061FDE8    n=0000000000061FDE0

a=0000000000061FE04    *a=1    b=1
p=0000000000061FE04    c=1    a=0000000000061FE04    b=16777216
x=0000000000061FDF4    *x=3.500000    y=3.500000
a=0000000000061FDF4    *a=3    x=0000000000061FDF4    *x=0.000000    y=0.000000
x=0000000000061FDF4    *x=12345.678711    y=12345.678711
p=0000000000061FDF4    c=-73    x=0000000000061FDF4    y=-0.000011
m=0000000000061FDE0    *m=-1.#QNAN0    n=-1.#QNAN0
m=0000000000061FDE0    *m=5.500000    n=5.500000
b=1    y=1.700000    n=1.700000
*a=1    *x=1.700000    *m=1.700000
n=1.700000    n=0000000000061FDE0    m=0000000000061FDF0
m=0000000000061FDF0    *m=0.300000

p=0000000000061FDF4    c=-73    a=0000000000061FE04    b=1    x=0000000000061FDF4    y=1.650000    m=0000000000061FDF0    n=1.700000

Process returned 0 (0x0)    execution time : 0.370 s
Press any key to continue.
```



второй вариант (m++;)

```

Enter b = 1
p=000000000061FE18      c=000000000061FE17      a=000000000061FE08      b=000000000061FE04      x=000000000061FDF8
      y=000000000061FDF4      m=000000000061FDE8      n=000000000061FDE0

a=000000000061FE04      *a=1      b=1
p=000000000061FE04      c=1      a=000000000061FE04      b=16777216
x=000000000061FDF4      *x=3.500000      y=3.500000
a=000000000061FDF4      *a=3      x=000000000061FDF4      *x=0.000000      y=0.000000
x=000000000061FDF4      *x=12345.678711      y=12345.678711
p=000000000061FDF4      c=-73      x=000000000061FDF4      y=-0.000011
m=000000000061FDE0      *m=-1.#QNAN0      n=-1.#QNAN0
m=000000000061FDE0      *m=5.500000      n=5.500000
b=1      y=1.700000      n=1.700000
*a=1      *x=1.700000      *m=1.700000
n=1.700000      n=000000000061FDE0      m=000000000061FDE8

Process returned -1073741819 (0xC0000005)   execution time : 3.099 s
Press any key to continue.

```

Размещение переменных в памяти:

адреса	переменные																																			
61fde0	n								m												y				x											
61fde1	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M					00	00	60	40	M	M	M	M	M	M	M						
61fde2																																				
61fde3																																				
61fde4																																				
61fde5																																				
61fde6																																				
61fde7																																				
61fde8																																				
61fde9																																				
61fdea																																				
61fdeb																																				
61fdec																																				
61fded																																				
61fdee																																				
61fdef																																				
61fdf0																																				
61fdf1																																				
61fdf2																																				
61fdf3																																				
61fdf4																																				
61fdf5																																				
61fdf6																																				
61fdf7																																				
61fdf8																																				
61fdf9																																				
61fdfa																																				
61fdfb																																				
61fdfc																																				
61fdfd																																				
61fdfe																																				
61fdff																																				
61fe00																																				
61fe01																																				
61fe02																																				
61fe03																																				
...																																				

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f		
	01	00	00	00	04	fe	61	00	00	00	00	00								м	м	м	м	м	м	м	м	м		
	b				a															c	p									
	↑a																													

```

p = (char*)a;
c = *p;
*p = *(p+3);
*(p+3) = c;
printf("p=%p\tc=%d\ta=%p\tb=%d\n", p, c, a, b);

```

Здесь указателю `p` присваивается значение указателя `a`, но так как указатель `char` на 1 байт, то `p` будет указывать только на этот байт. При этом мы меняем местами значения по адресам `61fe04` и `61fe07` с помощью переменной `c`, это удалось сделать, так смещение у `p` равно размеру `char` (1 байт), в переменной `b` получилось число  $1000000_{16} = 16777216_{10}$ .

адреса	переменные																																				
61fde0	n								m												y				x												
61fde1	m			m					m													00					m										
61fde2	m			m					m													00					m										
61fde3	m			m					m													60					m										
61fde4	m			m					m													40					m										
61fde5	m			m					m																		m										
61fde6	m			m					m																		m										
61fde7	m			m					m																		m										
61fde8	m			m					m																		m										
61fde9	m			m					m																		m										
61fdea	m			m					m																		m										
61fdeb	m			m					m																		m										
61fdec	m			m					m																		m										
61fded	m			m					m																		m										
61fdee	m			m					m																		m										
61fdef	m			m					m																		m										
61fdf0																																					
61fdf1																																					
61fdf2																																					
61fdf3																																					
61fdf4	00																																				
61fdf5	00																																				
61fdf6	60																																				
61fdf7	40																																				
61fdf8	m																																				
61fdf9	m																																				
61fdfa	m																																				
61fdfb	m																																				
61fdfc	m																																				
61fdfd	m																																				
61fdfe	m																																				
61fdff	m																																				
61fe00																																					
61fe01																																					
61fe02																																					
61fe03																																					

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f		
	00	00	00	01	04	fe	61	00	00	00	00	00								01	04	fe	61	00	00	00	00	00		
	b				a															c	p									
	↑ a																													
	↑ p																													

```

x = &y;
printf("x=%p\t*x=%f\ty=%f\n", x, *x, y);
a = (int*)x;
*a = *x;
printf("a=%p\t*a=%d\tx=%p\t*x=%f\ty=%f\n", a, *a, x, *x, y);

```

Здесь указателю x даётся адрес переменной y, а к указателю a присваивается значение указателя x, преобразованное в int. В результате изменения значения переменной y (\*a = \*x) в ней оказалось число 3, что в стандарте ieee754 будет очень маленьким числом, поэтому при выводе \*x (значения под указателем на float) получается 0. Напротив, при выводе \*a (значения под указателем на int) получается число, обрезанное до целого (3.5 (int) = 3), так получилось при присваивании \*a = \*x.

адреса																																
61fde0	m																															
61fde1	m																															
61fde2	m																															
61fde3	m																															
61fde4	m																															
61fde5	m																															
61fde6	m																															
61fde7	m																															
61fde8	m																															
61fde9	m																															
61fdea	m																															
61fdeb	m																															
61fdec	m																															
61fded	m																															
61fdee	m																															
61fdef	m																															
61fdf0																																
61fdf1																																
61fdf2																																
61fdf3																																
61fdf4	03																															
61fdf5	00																															
61fdf6	00																															
61fdf7	00																															
61fdf8	f4																															
61fdf9	fd																															
61fdfa	61																															
61fdfb	00																															
61fdfc	00																															
61fdfd	00																															
61fdfe	00																															
61fdff	00																															
61fe00																																
61fe01																																
61fe02																																
61fe03																																
...																																

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f		
	00	00	00	01	f4	fd	61	00	00	00	00	00								01	04	fe	61	00	00	00	00	00		
	b				a															c	p									
	↑ p																													

```

a = &b;
y = 12345.6789;
printf("x=%p\t*x=%f\ty=%f\n", x, *x, y);
p = (char*)x;
c=*p;
*p=*(p+3);
*(p+3)=c;
printf("p=%p\tc=%d\tx=%p\ty=%f\n", p, c, x, y);

```

Здесь присваивается указателю *p* значение указателя *x* с преобразованием в *char*, далее идёт обмен значениями по адресам *p* и *p+3* (61fdf4 и 61fdf7 соответственно), в результате при выводе переменной *y* получается очень маленькое число. т.к. смещенный порядок числа стал меньше ( $110 < 140$ ).

адреса	61fde0		61fde1		61fde2		61fde3		61fde4		61fde5		61fde6		61fde7		61fde8		61fde9		61fdea		61fdeb		61fdec		61fded		61fdee		61fdef		61fdf0		61fdf1		61fdf2		61fdf3		61fdf4		61fdf5		61fdf6		61fdf7		61fdf8		61fdf9		61fdfa		61fdfb		61fdfc		61fdfd		61fdfe		61fdff		61fe00		61fe01		61fe02		61fe03		...																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
	m		m		m		m		m		m		m		m		m		m		m		m		m		m		m		m		m		m		m		46		e6		40		b7		f4		fd		61		00		00		00		00		00		00		61fe00		61fe01		61fe02		61fe03																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
переменные	n																m																y																x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f									
	00	00	00	01	04	fe	61	00	00	00	00	00								b7	f4	fd	61	00	00	00	00	00									
	b				a															c	p																
	↑ a																																				
	↑ p																																				

```
m = &n;
printf("m=%p\t*m=%lf\tn=%lf\n", m, *m, n);
n = 5.5;
```

```
printf("m=%p\t*m=%lf\tn=%lf\n", m, *m, n);
b = n = y = 1.7;
printf("b=%d\ty=%f\tn=%lf\n", b, y, n);
printf("a=%d\t*x=%f\t*m=%lf\n", *a, *x, *m);
m+=2;
printf("n=%lf\tn=%p\tn=%p\n", n, &n, m);
*m = (float)*a - n + (int)*x;
printf("m=%p\t*m=%lf\n\n", m, *m);
printf("p=%p\ta=%d\tb=%d\tc=%d\tx=%p\ty=%f\tn=%f\n\n", p, c, a, b, x, y, m, n);
```

при  $m+=2$  указатель был смещён на 16 байт ( $2 \cdot \text{sizeof}(\text{double})$ ), первые 4 ячейки не являются какой-либо переменной, а вторые 4 ячейки — это переменная  $y$ . Потом вычисляется значение:

$*m = (\text{float}) *a - n + (\text{int}) *x$ :

$*m = 1. - 1.7 + 1 = 0.3$

При выводе  $y$  отображается 1.650000, так как на порядок  $y$  `float` отводится меньше, чем у `double`, в итоге сам порядок в представлении `float` стал больше, и число изменилось ( $0.300000 \Rightarrow 1.650000$ ).

адреса	61fde0	61fde1	61fde2	61fde3	61fde4	61fde5	61fde6	61fde7	61fde8	61fde9	61fdea	61fdeb	61fdec	61fded	61fdee	61fdef	61fdf0	61fdf1	61fdf2	61fdf3	61fdf4	61fdf5	61fdf6	61fdf7	61fdf8	61fdf9	61fdfa	61fdfb	61fdfc	61fdfd	61fdfe	61fdff	61fe00	61fe01	61fe02	61fe03	...
	33	33	33	33	33	33	fb	3f	f0	fd	61	00	00	00	00	00	33	33	33	33	33	33	d3	3f	f4	fd	61	00	00	00	00	00					
переменные	n								m												y				x												
																	↑ m																				
																					↑ x																

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f		
	01	00	00	00	04	fe	61	00	00	00	00	00								b7	f4	fd	61	00	00	00	00	00		
	b				a															c	p									
	↑ a																													
	↑ p																													

После замены инструкции «m+=2;» инструкцией «m++;» указатель стал ссылаться на себя. При попытке изменить значение под указателем выводится ошибка, так как это значение (адрес) будет находиться в другой, недоступной для компилятора, области памяти. (segmentation fault – ошибка доступа к памяти).

адреса																																				
61fde0	33																																			
61fde1	33																																			
61fde2	33																																			
61fde3	33																																			
61fde4	33																																			
61fde5	33																																			
61fde6	fb																																			
61fde7	3f																																			
61fde8	e8																																			
61fde9	fd																																			
61fdea	61																																			
61fdeb	00																																			
61fdec	00																																			
61fded	00																																			
61fdee	00																																			
61fdef	00																																			
61fdf0																																				
61fdf1																																				
61fdf2																																				
61fdf3																																				
61fdf4	9a																																			
61fdf5	99																																			
61fdf6	d9																																			
61fdf7	3f																																			
61fdf8	f4																																			
61fdf9	fd																																			
61fdfa	61																																			
61fdfb	00																																			
61fdfc	00																																			
61fdfd	00																																			
61fdfe	00																																			
61fdff	00																																			
61fe00																																				
61fe01																																				
61fe02																																				
61fe03																																				
...																																				
переменные	n								m												y				x											
									↑ m												↑ x															

...	61fe04	61fe05	61fe06	61fe07	61fe08	61fe09	61fe0a	61fe0b	61fe0c	61fe0d	61fe0e	61fe0f	61fe10	61fe11	61fe12	61fe13	61fe14	61fe15	61fe16	61fe17	61fe18	61fe19	61fe1a	61fe1b	61fe1c	61fe1d	61fe1e	61fe1f
	00	00	00	01	04	fe	61	00	00	00	00	00								b7	f4	fd	61	00	00	00	00	00
	b				a															c	p							
	↑ a																											
	↑ p								↑ p																			

### Задание 3.

Объявить по две переменные типов *char*, *int* и *double*, а также указатель на *char*. Вывести на экран размеры и адреса всех переменных, начертить схему расположения переменных в памяти. Поменять порядок объявления переменных (например, *int*, *char*, *double*, *char\**, *char*, *double*, *int*). Запустить программу повторно, проанализировать, что изменилось. Задать переменной типа *int* такое значение, чтобы значение каждого байта было уникальным, использовать для этого шестнадцатеричную константу. Записать адрес этой переменной в указатель на *char* и с его помощью вывести на экран содержимое каждого байта (тоже в шестнадцатеричной системе счисления). Проанализировать, прямой или обратный порядок расположения байт при записи числа применяется в используемой системе.

Повторить выполнение этого задания на компьютере с другой операционной системой и/или другой IDE (можно использовать домашний компьютер или онлайн-компилятор). Сравнить результаты работы программы на разных платформах, сделать выводы.

#### 1 Операционная система KDE neon (Ubuntu 20.04.2 LTS), среда разработки Code::Blocks

Текст программы (первый порядок объявления переменных):

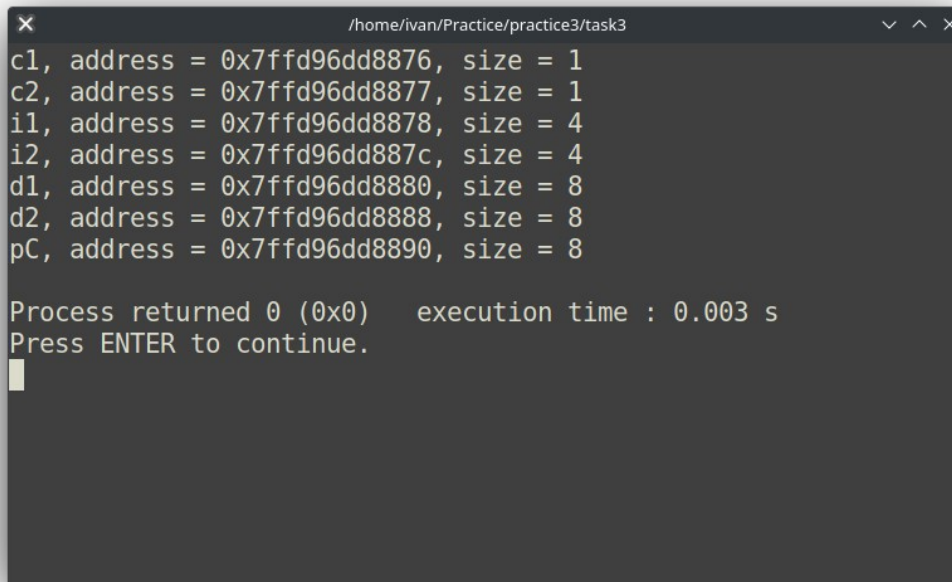
```
#include <stdio.h>

int main(){
    char c1,c2;
    int i1,i2;
    double d1,d2;
```



```
char* pC;  
printf("c1, address = %p, size = %d\n",&c1,sizeof(c1));  
printf("c2, address = %p, size = %d\n",&c2,sizeof(c2));  
printf("i1, address = %p, size = %d\n",&i1,sizeof(i1));  
printf("i2, address = %p, size = %d\n",&i2,sizeof(i2));  
printf("d1, address = %p, size = %d\n",&d1,sizeof(d1));  
printf("d2, address = %p, size = %d\n",&d2,sizeof(d2));  
printf("pC, address = %p, size = %d\n",&pC,sizeof(pC));  
return 0;  
}
```

Результаты работы программы:



```
x /home/ivan/Practice/practice3/task3  
c1, address = 0x7ffd96dd8876, size = 1  
c2, address = 0x7ffd96dd8877, size = 1  
i1, address = 0x7ffd96dd8878, size = 4  
i2, address = 0x7ffd96dd887c, size = 4  
d1, address = 0x7ffd96dd8880, size = 8  
d2, address = 0x7ffd96dd8888, size = 8  
pC, address = 0x7ffd96dd8890, size = 8  
  
Process returned 0 (0x0)   execution time : 0.003 s  
Press ENTER to continue.  
█
```

### Размещение переменных в памяти:

адреса	dd8876	dd8877	dd8878	dd7879	dd787a	dd787b	dd787c	dd787d	dd787e	dd787f	dd7880	dd7881	dd7882	dd7883	dd7884	dd7885	dd7886	dd7887	dd7888	dd7889	dd788a	dd788b	dd788c	dd788d	dd788e	dd788f	dd7890	dd7891	dd7892	dd7893	dd7894	dd7895	dd7896	dd7897			
переменные	c1	c2	i1				i2				d1									d2							pC										

### Текст программы (второй порядок объявления переменных):

```
#include <stdio.h>

int main(){
    char c1;
    double d1;
    char *pC;
    int i1;
    double d2;
    char c2;
    int i2 = 0x7f5f3a22;
    printf("c1, address = %p, size = %d\n",&c1,sizeof(c1));
    printf("c2, address = %p, size = %d\n",&c2,sizeof(c2));
    printf("i1, address = %p, size = %d\n",&i1,sizeof(i1));
    printf("i2, address = %p, size = %d\n",&i2,sizeof(i2));
    printf("d1, address = %p, size = %d\n",&d1,sizeof(d1));
    printf("d2, address = %p, size = %d\n",&d2,sizeof(d2));
    printf("pC, address = %p, size = %d\n\n",&pC,sizeof(pC));
    printf("i2 value = %x\n",i2);
    printf("i2 byte structure:\n\n");
    for(pC = (char*)&i2; pC < (char*)&i2+sizeof(int); pC++){
        printf("%x\n",*pC);
    }
    return 0;
}
```

}

Результаты работы программы:

```
/home/ivan/Practice/practice3/task3B
c1, address = 0x7ffce13317d6, size = 1
c2, address = 0x7ffce13317d7, size = 1
i1, address = 0x7ffce13317d8, size = 4
i2, address = 0x7ffce13317dc, size = 4
d1, address = 0x7ffce13317e0, size = 8
d2, address = 0x7ffce13317f0, size = 8
pC, address = 0x7ffce13317e8, size = 8

i2 value = 7f5f3a22
i2 byte structure:

22
3a
5f
7f

Process returned 0 (0x0)   execution time : 0.004 s
Press ENTER to continue.
```

Размещение переменных в памяти:

переменные	адреса																											
c1	3317d6																											
	3317d7																											
c2	3317d8																											
	3317d9																											
i1	3317da																											
	3317db																											
i2	22																											
	3a																											
d1	5f																											
	7f																											
d1	3317e0																											
	3317e1																											
d1	3317e2																											
	3317e3																											
d1	3317e4																											
	3317e5																											
d1	3317e6																											
	3317e7																											
pC	df																											
	17																											
pC	33																											
	e1																											
pC	fc																											
	7f																											
pC	00																											
	00																											
d2	3317f0																											
	3317f1																											
d2	3317f2																											
	3317f3																											
d2	3317f4																											
	3317f5																											
d2	3317f6																											
	3317f7																											

*Выводы:* память выделяется по возрастанию размера переменных в байтах (т.е. char (1 байт), int (4 байт), double и указатели (8 байт)).

Порядок следования байт при записи числа - *обратный*.

2 Операционная система KDE neon (Ubuntu 20.04.2 LTS), среда разработки онлайн-компилятор Replit <https://replit.com/languages/c>

Результаты работы программы при первом порядке объявления переменных:

```
c1, address = 0x7ffd43ccc5fb, size = 1
c2, address = 0x7ffd43ccc5fa, size = 1
i1, address = 0x7ffd43ccc5f4, size = 4
i2, address = 0x7ffd43ccc5f0, size = 4
d1, address = 0x7ffd43ccc5e8, size = 8
d2, address = 0x7ffd43ccc5e0, size = 8
pC, address = 0x7ffd43ccc5d8, size = 8
```

Размещение переменных в памяти:

адреса	ccc5d8	ccc5d9	ccc5da	ccc5db	ccc5dc	ccc5dd	ccc5de	ccc5df	ccc5e0	ccc5e1	ccc5e2	ccc5e3	ccc5e4	ccc5e5	ccc5e6	ccc5e7	ccc5e8	ccc5e9	ccc5ea	ccc5eb	ccc5ec	ccc5ed	ccc5ee	ccc5ef	ccc5f0	ccc5f1	ccc5f2	ccc5f3	ccc5f4	ccc5f5	ccc5f6	ccc5f7	ccc5f8	ccc5f9	ccc5fa	ccc5fb
переменные	pC								d2								d1								i2				i1						c2	c1

Результаты работы программы при втором порядке объявления переменных:

```
c1, address = 0x7ffdb015e23b, size = 1
c2, address = 0x7ffdb015e217, size = 1
i1, address = 0x7ffdb015e224, size = 4
i2, address = 0x7ffdb015e210, size = 4
d1, address = 0x7ffdb015e230, size = 8
d2, address = 0x7ffdb015e218, size = 8
pC, address = 0x7ffdb015e228, size = 8

i2 value = 7f5f3a22
i2 byte structure:

22
3a
5f
7f
```

Размещение переменных в памяти:

переменные	адреса
i2	22
	3a
	5f
	7f
c2	
d2	
i1	
pC	13
	e2
	15
	b0
	fd
	7f
	00
	00
d1	
c1	

*Выводы:* память под переменные выделяется в том порядке, в котором они были объявлены, выделение памяти идёт от большего адреса к меньшему.

Порядок следования байт при записи числа также *обратный*.

*Выводы:* Распределение памяти под переменные зависит от компилятора (gcc или clang) и ОС (Windows или Linux), но в обеих системах порядок следования байт при записи числа является *обратным*.