

# Задание 4

VARIABLES

Locals

d = 18.123434339999999

pB = 0x5ffe2b

\*pB = true

b = true

pC = 0x5ffe2a "q\001\177"

\*pC = 113 'q'

pI = 0x5ffe24

\*pI = 100000

pf = 0x5ffe20

\*pf = 17

c = 113 'q'

s = 12000

i = 100000

f = 17

WATCH

hw4 > lab > lab2.cpp > main()

21 int main()

25 //Задание 4. Указатели

26 //Объявите объекты разных типов и проинициализируйте их.

27 //Объявите указатели соответствующих типов и проинициализируйте их

28 //адресами заданных объектов.

29 //Выполняя задание, с помощью отладчика посмотрите, какие значения

30 //будут присвоены переменным-указателям и на что эти указатели

31 //указывают

32 {

33 bool b = true;

34 char c = 'q';

35 short s = 12000;

36 int i = 100000;

37 float f = 17.0;

38 double d = 18.12343434;

39

40 bool \*pB = &b; // true

41 char \*pC = &c; // 113 'q'

42 short \*pS = &s; // 12000

43 int \*pI = &i; // 100000

44 float \*pf = &f; // 17

45 double \*pD = &d; // 18.12343434

46 }

VARIABLES

Locals

pC = 0x5ffe2a "q\001\177"

pI = 0x5ffe24

\*pI = 100000

pf = 0x5ffe20

\*pf = 17

c = 113 'q'

s = 12000

i = 100000

f = 17

pS = 0x5ffe28

\*pS = 12000

pD = 0x1914b0

\*pD = 8.1212534600800742e-318

Registers

WATCH

hw4 > lab > lab2.cpp > main()

21 int main()

25 //Задание 4. Указатели

26 //Объявите объекты разных типов и проинициализируйте их.

27 //Объявите указатели соответствующих типов и проинициализируйте их

28 //адресами заданных объектов.

29 //Выполняя задание, с помощью отладчика посмотрите, какие значения

30 //будут присвоены переменным-указателям и на что эти указатели

31 //указывают

32 {

33 bool b = true;

34 char c = 'q';

35 short s = 12000;

36 int i = 100000;

37 float f = 17.0;

38 double d = 18.12343434;

39

40 bool \*pB = &b; // true

41 char \*pC = &c; // 113 'q'

42 short \*pS = &s; // 12000

43 int \*pI = &i; // 100000

44 float \*pf = &f; // 17

45 double \*pD = &d; // 8.1212534600800742e-318

46 }

# Задание 4а

VARIABLES

Locals

str = 0x7fff6e038b000 "Hello"

c = 101 'e'

Registers

WATCH

hw4 > lab > lab2.cpp > main()

21 int main()

49

50

51 //Задание 4а. объявите указатель на тип char и проинициализируйте

52 //его адресом строкового литерала. Присвойте переменной типа char значение

53 //любого элемента строки.

54 //Проверьте - возможно ли присвоить какое-либо значение по адресу,

55 //задаваемому указателем?

56 {

57 const char\* str = "Hello";

58 char c = str[1]; // содержит 2-й элемент (счет с 0-го) строкового литерала str - 101 'e'

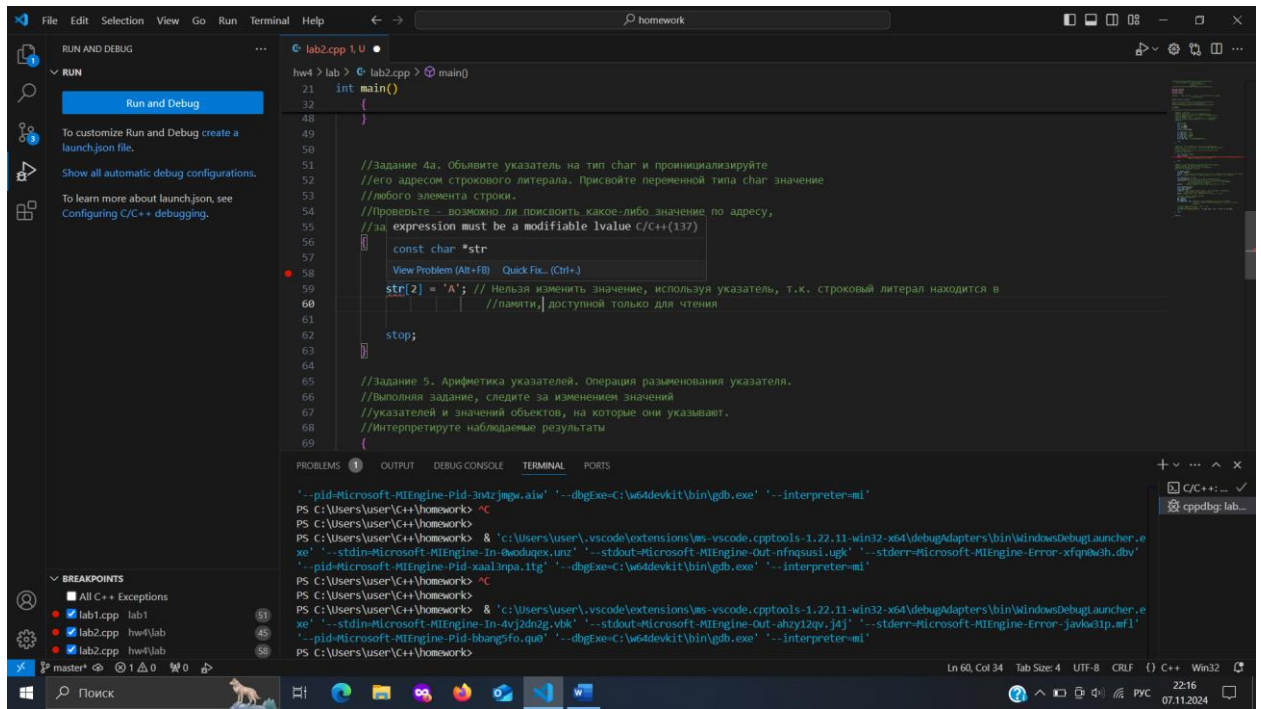
59

60 //str[2] = 'A'; Нельзя изменить значение, используя указатель, т.к. строковый литерал находится в

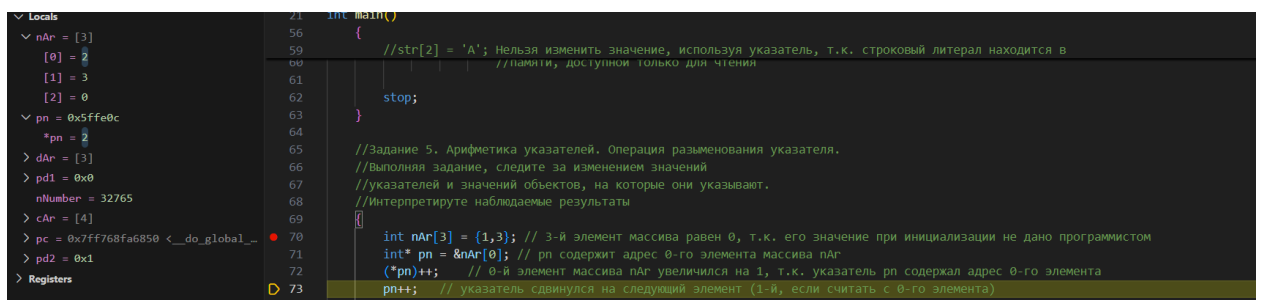
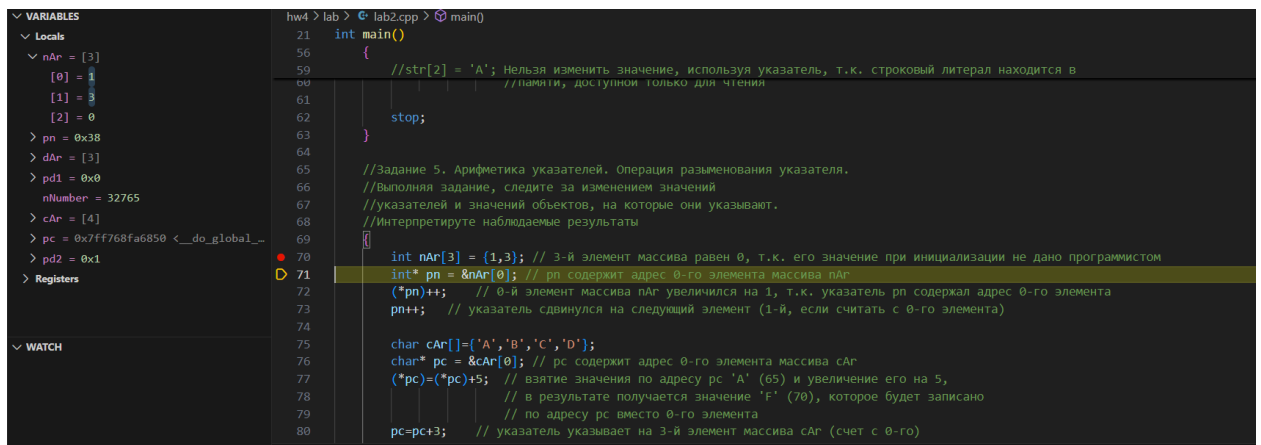
61 //памяти, доступной только для чтения

62

63 stop;



## Задание 5



```
Locals
nAr = [3]
[0] = 2
[1] = 3
[2] = 0
pn = 0x5ffe10
*pn = 3
dAr = [3]
pd1 = 0x0
nNumber = 32765
cAr = [4]
pc = 0x7ff768fa6850 <__do_global_...
pd2 = 0x1
Registers
WATCH
21 int main()
22 {
56 //Задание 5. Арифметика указателей. Операция разыменования указателя.
59 //Выполняя задание, следите за изменением значений
60 //указателей и значений объектов, на которые они указывают.
61 //Интерпретируйте наблюдаемые результаты
62 stop;
63 }
64
65 //Задание 5. Арифметика указателей. Операция разыменования указателя.
66 //Выполняя задание, следите за изменением значений
67 //указателей и значений объектов, на которые они указывают.
68 //Интерпретируйте наблюдаемые результаты
69
70 int nAr[3] = {1,3}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
71 int* pn = &nAr[0]; // pn содержит адрес 0-го элемента массива nAr
72 (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержит адрес 0-го элемента
73 pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75 char cAr[]={'A','B','C','D'};
```

```
VARIABLES
Locals
nAr = [3]
pn = 0x5ffe10
dAr = [3]
pd1 = 0x0
nNumber = 32765
cAr = [4]
[0] = 65 'A'
[1] = 66 'B'
[2] = 67 'C'
[3] = 68 'D'
pc = 0x5ffe08 "ABCD\002"
*pc = 65 'A'
pd2 = 0x1
Registers
hw4 > lab > lab2.cpp > main()
21 int main()
22 {
65 //Задание 5. Арифметика указателей. Операция разыменования указателя.
67 //указателей и значений объектов, на которые они указывают.
68 //Интерпретируйте наблюдаемые результаты
69
70 int nAr[3] = {1,3}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
71 int* pn = &nAr[0]; // pn содержит адрес 0-го элемента массива nAr
72 (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержит адрес 0-го элемента
73 pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75 char cAr[]={'A','B','C','D'};
76 char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77 (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78 // в результате получается значение 'F' (70), которое будет записано
79 // по адресу pc вместо 0-го элемента
80 pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82 double dAr[3]={1.0,2.0}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
```

```
Locals
nAr = [3]
pn = 0x5ffe10
dAr = [3]
pd1 = 0x0
nNumber = 32765
cAr = [4]
[0] = 70 'F'
[1] = 66 'B'
[2] = 67 'C'
[3] = 68 'D'
pc = 0x5ffe08 "ABCD\002"
*pc = 70 'F'
pd2 = 0x1
Registers
21 int main()
22 {
65 //Задание 5. Арифметика указателей. Операция разыменования указателя.
67 //указателей и значений объектов, на которые они указывают.
68 //Интерпретируйте наблюдаемые результаты
69
70 int nAr[3] = {1,3}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
71 int* pn = &nAr[0]; // pn содержит адрес 0-го элемента массива nAr
72 (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержит адрес 0-го элемента
73 pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75 char cAr[]={'A','B','C','D'};
76 char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77 (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78 // в результате получается значение 'F' (70), которое будет записано
79 // по адресу pc вместо 0-го элемента
80 pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82 double dAr[3]={1.0,2.0}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
```

```
Locals
nAr = [3]
pn = 0x5ffe10
dAr = [3]
pd1 = 0x0
nNumber = 32765
cAr = [4]
[0] = 70 'F'
[1] = 66 'B'
[2] = 67 'C'
[3] = 68 'D'
pc = 0x5ffe0b "D\002"
*pc = 68 'D'
pd2 = 0x1
Registers
21 int main()
22 {
65 //Задание 5. Арифметика указателей. Операция разыменования указателя.
67 //указателей и значений объектов, на которые они указывают.
68 //Интерпретируйте наблюдаемые результаты
69
70 int nAr[3] = {1,3}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
71 int* pn = &nAr[0]; // pn содержит адрес 0-го элемента массива nAr
72 (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержит адрес 0-го элемента
73 pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75 char cAr[]={'A','B','C','D'};
76 char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77 (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78 // в результате получается значение 'F' (70), которое будет записано
79 // по адресу pc вместо 0-го элемента
80 pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82 double dAr[3]={1.0,2.0}; // 3-й элемент массива равен 0, т.к. его значение при инициализации не дано программистом
```

```
Locals
nAr = [3]
pn = 0x5ffe10
dAr = [3]
[0] = 1
[1] = 2
[2] = 0
pd1 = 0x0
nNumber = 32765
cAr = [4]
pc = 0x5ffe0b "D\002"
pd2 = 0x1
Registers
21 int main()
22 {
69 {
72 (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержит адрес 0-го элемента
73 pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75 char cAr[]={'A','B','C','D'};
76 char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77 (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78 // в результате получается значение 'F' (70), которое будет записано
79 // по адресу pc вместо 0-го элемента
80 pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82 double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83 double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84 double* pd2 = pd1;
85 (*pd2)+2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86 // значения 0-го элемента изменится на 3-й.
```

```

VARIABLES
└─ Locals
  > nAr = [3]
  > pn = 0x5ffe10
  > dAr = [3]
    [0] = 1
    [1] = 2
    [2] = 0
  > pd1 = 0x5ffdf0
    *pd1 = 1
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x1
  > Registers

21 int main()
69 {
72     (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержал адрес 0-го элемента
73     pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75     char cAr[]={'A','B','C','D'};
76     char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77     (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78                     // в результате получается значение 'F' (70), которое будет записано
79                     // по адресу pc вместо 0-го элемента
80     pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.

```

```

Locals
└─ > pn = 0x5ffe10
  > dAr = [3]
    [0] = 1
    [1] = 2
    [2] = 0
  > pd1 = 0x5ffdf0
    *pd1 = 1
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x5ffdf0
    *pd2 = 1
  > Registers

21 int main()
69 {
72     (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержал адрес 0-го элемента
73     pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75     char cAr[]={'A','B','C','D'};
76     char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77     (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78                     // в результате получается значение 'F' (70), которое будет записано
79                     // по адресу pc вместо 0-го элемента
80     pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.

```

```

Locals
└─ > pn = 0x5ffe10
  > dAr = [3]
    [0] = 3
    [1] = 2
    [2] = 0
  > pd1 = 0x5ffdf0
    *pd1 = 3
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x5ffdf0
    *pd2 = 3
  > Registers

21 int main()
69 {
72     (*pn)++; // 0-й элемент массива nAr увеличился на 1, т.к. указатель pn содержал адрес 0-го элемента
73     pn++; // указатель сдвинулся на следующий элемент (1-й, если считать с 0-го элемента)
74
75     char cAr[]={'A','B','C','D'};
76     char* pc = &cAr[0]; // pc содержит адрес 0-го элемента массива cAr
77     (*pc)=(*pc)+5; // взятие значения по адресу pc 'A' (65) и увеличение его на 5,
78                     // в результате получается значение 'F' (70), которое будет записано
79                     // по адресу pc вместо 0-го элемента
80     pc=pc+3; // указатель указывает на 3-й элемент массива cAr (счет с 0-го)
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.
87     pd2+=2; // указатель pd2, теперь указывает на 2-й элемент (счет с 0-го)

```

```

Locals
└─ > pn = 0x5ffe10
  > dAr = [3]
    [0] = 3
    [1] = 2
    [2] = 0
  > pd1 = 0x5ffdf0
    *pd1 = 3
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x5ffe00
    *pd2 = 0
  > Registers

69 {
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.
87     pd2+=2; // указатель pd2, теперь указывает на 2-й элемент (счет с 0-го)
88
89     //Объясните результат выполнения операции
90     //вычитания двух указателей
91     pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
92     pd2 = &dAr[1]; // pd2 содержит адрес 1-го элемента массива dAr
93     int nNumber=pd2 - pd1; // операция вычитания указателей имеет смысл для определения
94                         // количества элементов между двумя указателями в рамках одного массива
95

```

```

Locals
└─ > nAr = [3]
  > pn = 0x5ffe10
  > dAr = [3]
  > pd1 = 0x5ffdf0
    *pd1 = 3
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x5ffe00
  > Registers

21 int main()
69 {
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.
87     pd2+=2; // указатель pd2, теперь указывает на 2-й элемент (счет с 0-го)
88
89     //Объясните результат выполнения операции
90     //вычитания двух указателей
91     pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
92     pd2 = &dAr[1]; // pd2 содержит адрес 1-го элемента массива dAr
93     int nNumber=pd2 - pd1; // операция вычитания указателей имеет смысл для определения
94                         // количества элементов между двумя указателями в рамках одного массива
95

```

```

Locals
└─ > nAr = [3]
  > pn = 0x5ffe10
  > dAr = [3]
  > pd1 = 0x5ffdf0
    *pd1 = 3
    nNumber = 32765
  > cAr = [4]
  > pc = 0x5ffe0b "D\002"
  > pd2 = 0x5ffdf0
    *pd2 = 2
  > Registers

21 int main()
69 {
81
82     double dAr[3]={1.0,2.0}; // 2-й элемент массива (счет с 0-го) равен 0, т.к. его значение при инициализации не дано программистом
83     double* pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
84     double* pd2 = pd1;
85     (*pd2)+=2; // pd2 - указывает на 0-й элемент массива dAr. После разыменования и прибавления 2,
86                 // значение 0-го элемента изменится на 3.0.
87     pd2+=2; // указатель pd2, теперь указывает на 2-й элемент (счет с 0-го)
88
89     //Объясните результат выполнения операции
90     //вычитания двух указателей
91     pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
92     pd2 = &dAr[1]; // pd2 содержит адрес 1-го элемента массива dAr
93     int nNumber=pd2 - pd1; // операция вычитания указателей имеет смысл для определения
94                         // количества элементов между двумя указателями в рамках одного массива
95

```

```
Locals
> nAr = [3]
> pn = 0x5ffe10
> dAr = [3]
  pd1 = 0x5ffdf0
    *pd1 = 3
  nNumber = 1
> cAr = [4]
> pc = 0x5ffe0b "D:\002"
  pd2 = 0x5ffdf8
    *pd2 = 2
Registers

91 int main()
92 {
93     //Обьясните результат выполнения операции
94     //вычитания двух указателей
95     pd1 = &dAr[0]; // pd1 содержит адрес 0-го элемента массива dAr
96     pd2 = &dAr[1]; // pd2 содержит адрес 1-го элемента массива dAr
97     int nNumber=pd2 - pd1; // операция вычитания указателей имеет смысл для определения
98                           // количества элементов между двумя указателями в рамках одного массива
99     // nNumber = 1
100     //Сравните указатели pd2 и pd1 и с помощью cout
101     //выведите результаты сравнения.
102     std::cout << "5. Compare result is: " << (pd2 > pd1 ? "true" : "false") << std::endl;
103
104     stop;
105 }
```

```
99     std::cout << "5. Compare result is: " << (pd2 > pd1 ? "true" : "false") << std::endl; // true
100
101     stop;
102 }
103
104     return 0;
105 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\user\C++\homework> ^C
PS C:\Users\user\C++\homework>
PS C:\Users\user\C++\homework> & 'c:\Users\user\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-qsb0q0pd.v4p' '--stdout=Microsoft-MIEngine-Out-ktir0xq0.wis' '--stderr=Microsoft-MIEngine-Error-pelashkz.l35' '--pid=Microsoft-MIEngine-Pid-odcrd1ig.wrx' '--dbgExe=c:\w64devkit\bin\gdb.exe' '--interpreter=mi'
5. Compare result is: true
```