

ИДЗ №1 (Отчет на 5)

Воробьев Артём Александрович БПИ214 Вариант №10

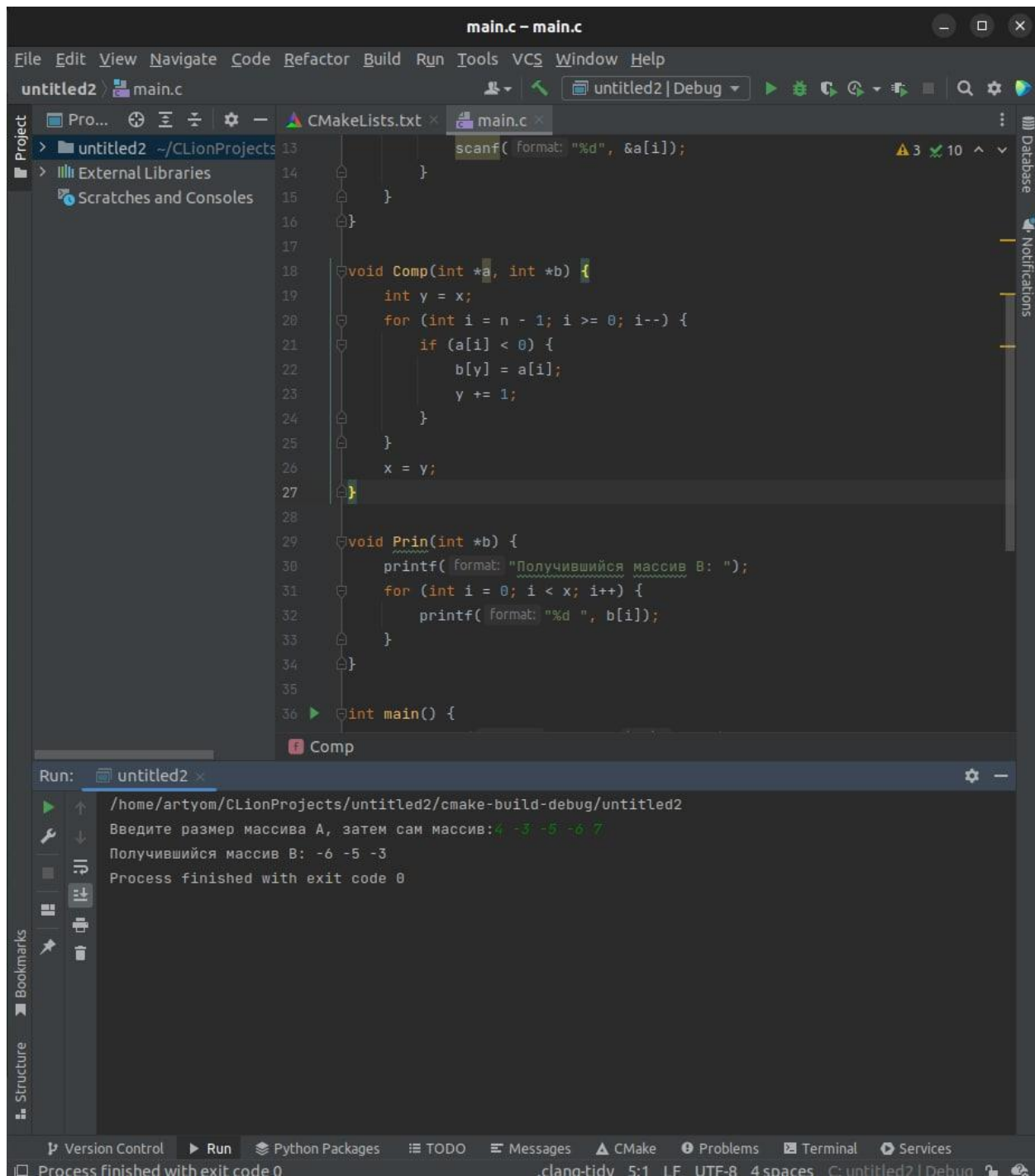
Условие задания: сформировать массив В из отрицательных элементов массива А, расположенных обратном порядке.

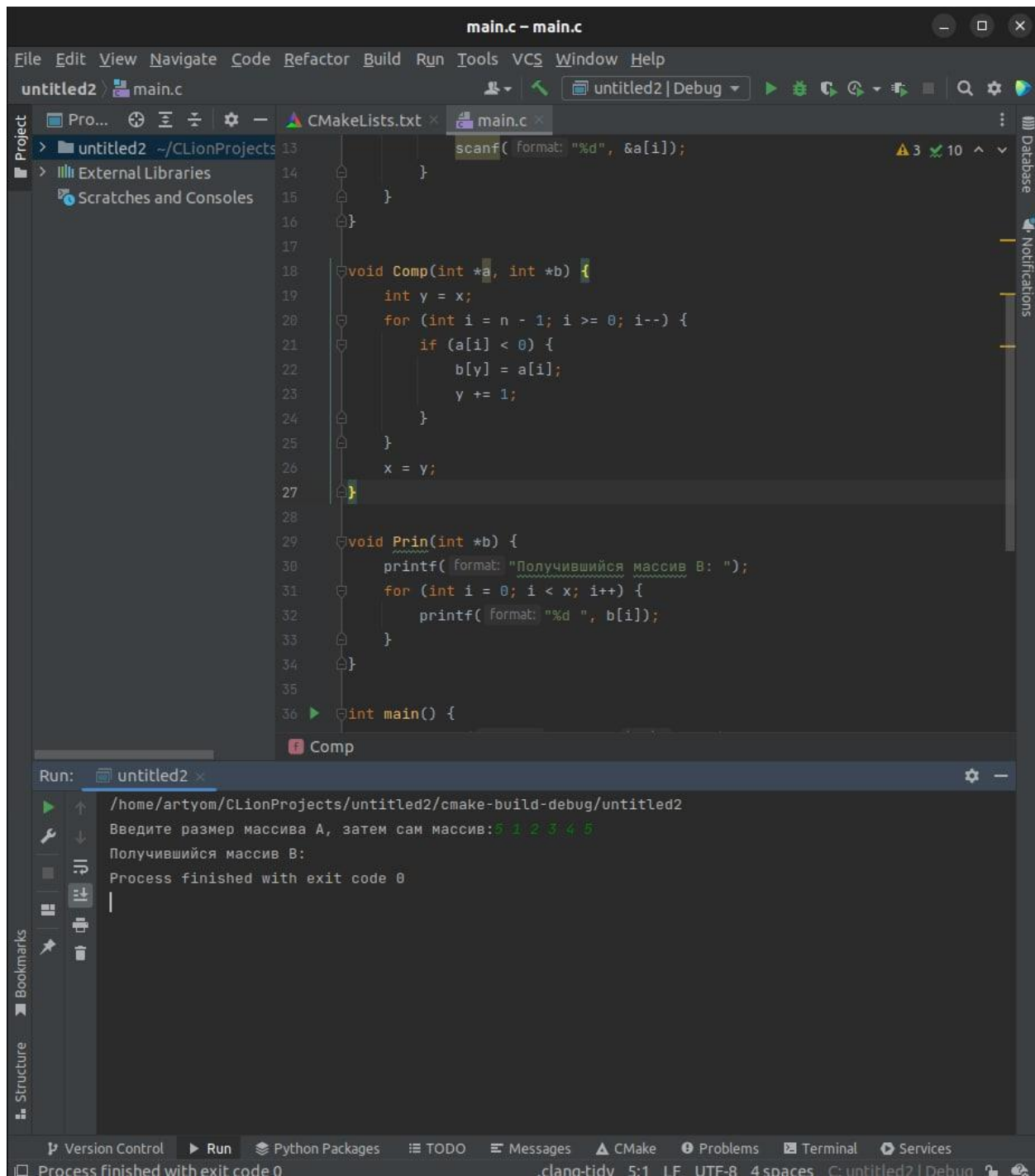
- 1) Тесты для проверки работы программ на Си и Ассемблере:

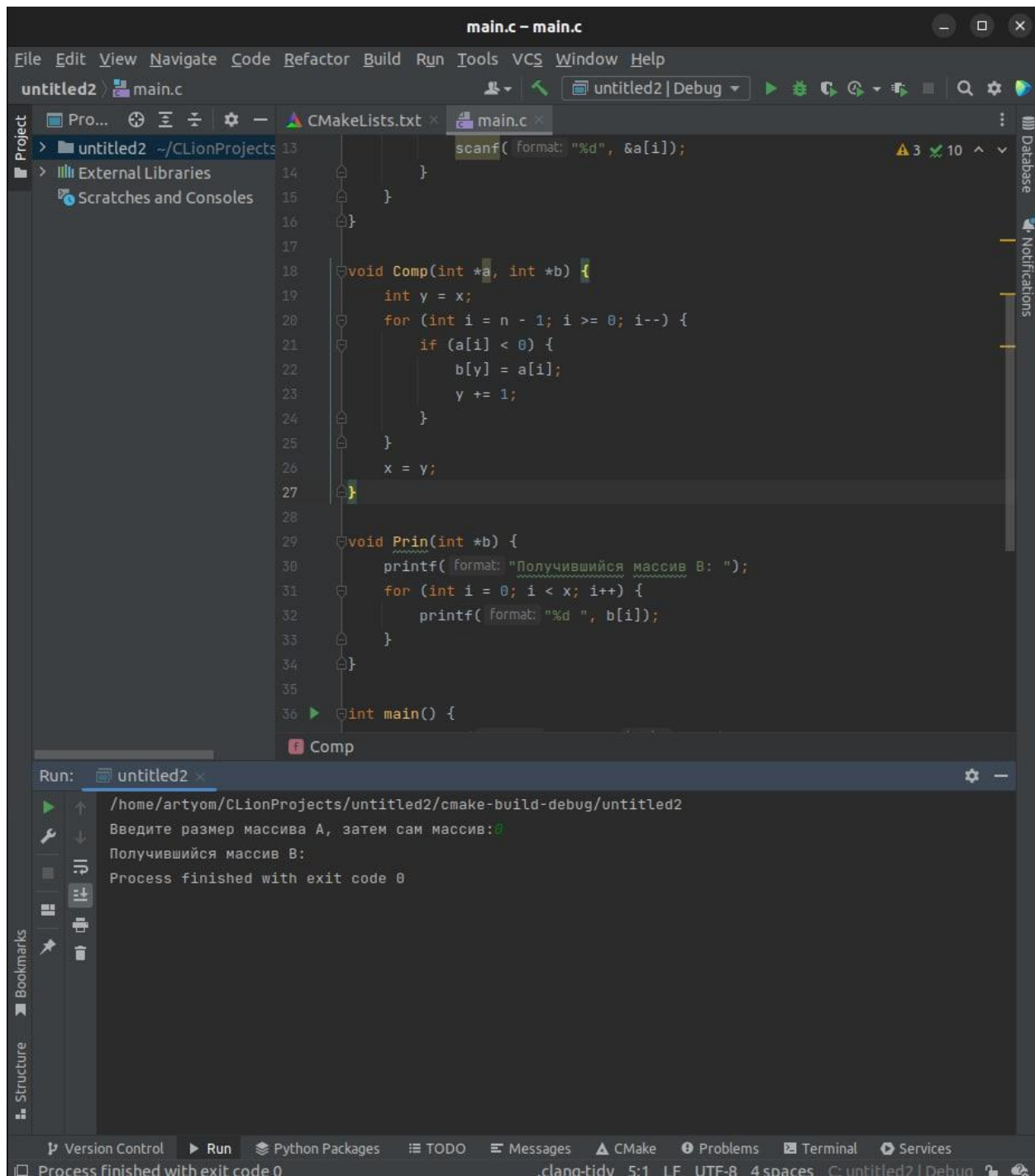
```
Терминал
Пт, 21 октября 18:22
artyom@artyom-GL703VM: ~/CLionProjects/untitled2

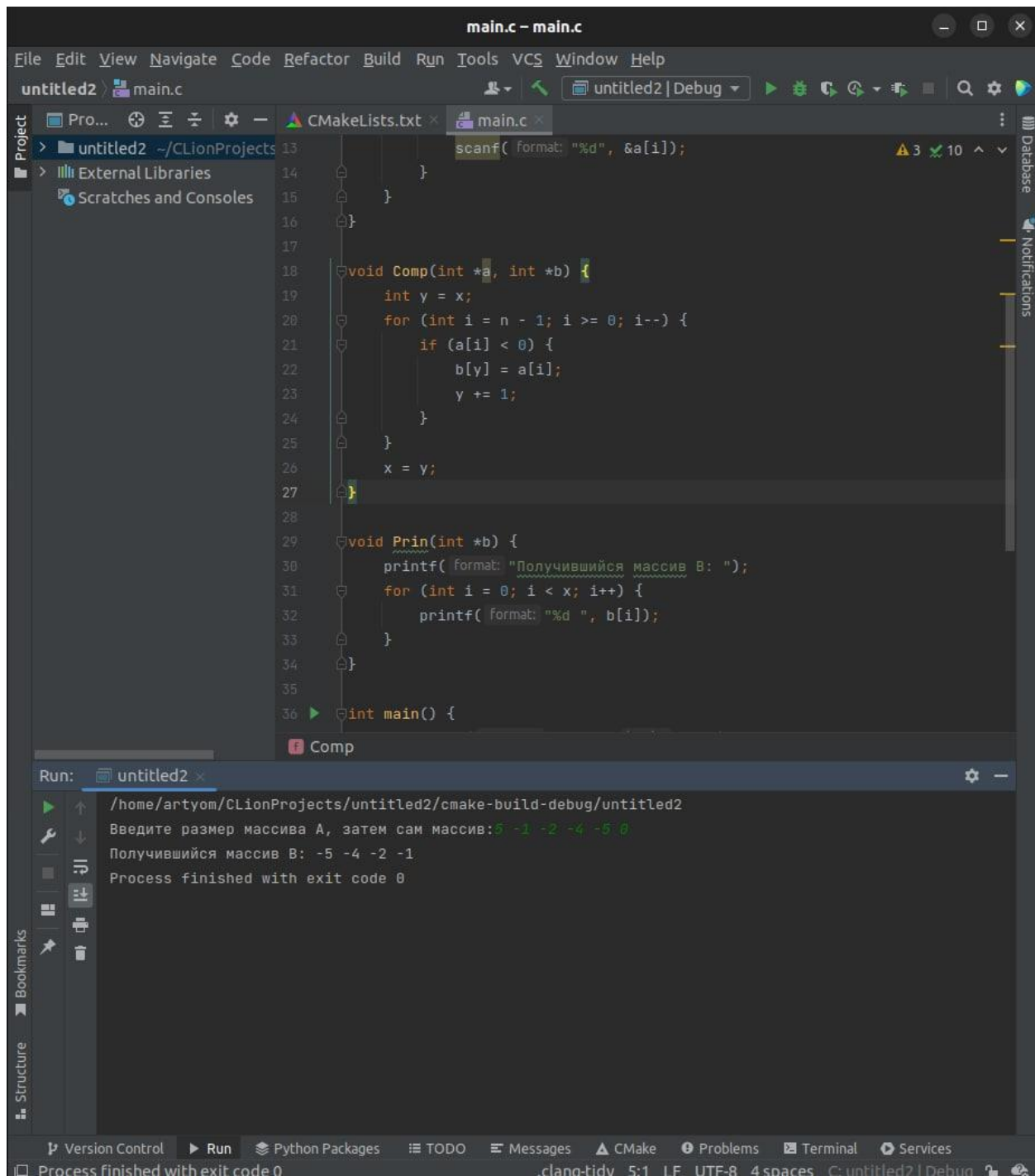
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./main
Введите размер массива А, затем сам массив:4 -3 -5 -6 7
Получившийся массив В: -6 -5 -3
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./main
Введите размер массива А, затем сам массив:5 1 2 3 4 5
Получившийся массив В:
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./main
Введите размер массива А, затем сам массив:0
Получившийся массив В:
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./main
Введите размер массива А, затем сам массив:5
-1 -2 -4 -5 0
Получившийся массив В: -5 -4 -2 -1
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./main
Введите размер массива А, затем сам массив:5
-100 -99 -98 -97 -96
Получившийся массив В: -96 -97 -98 -99 -100
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ 5

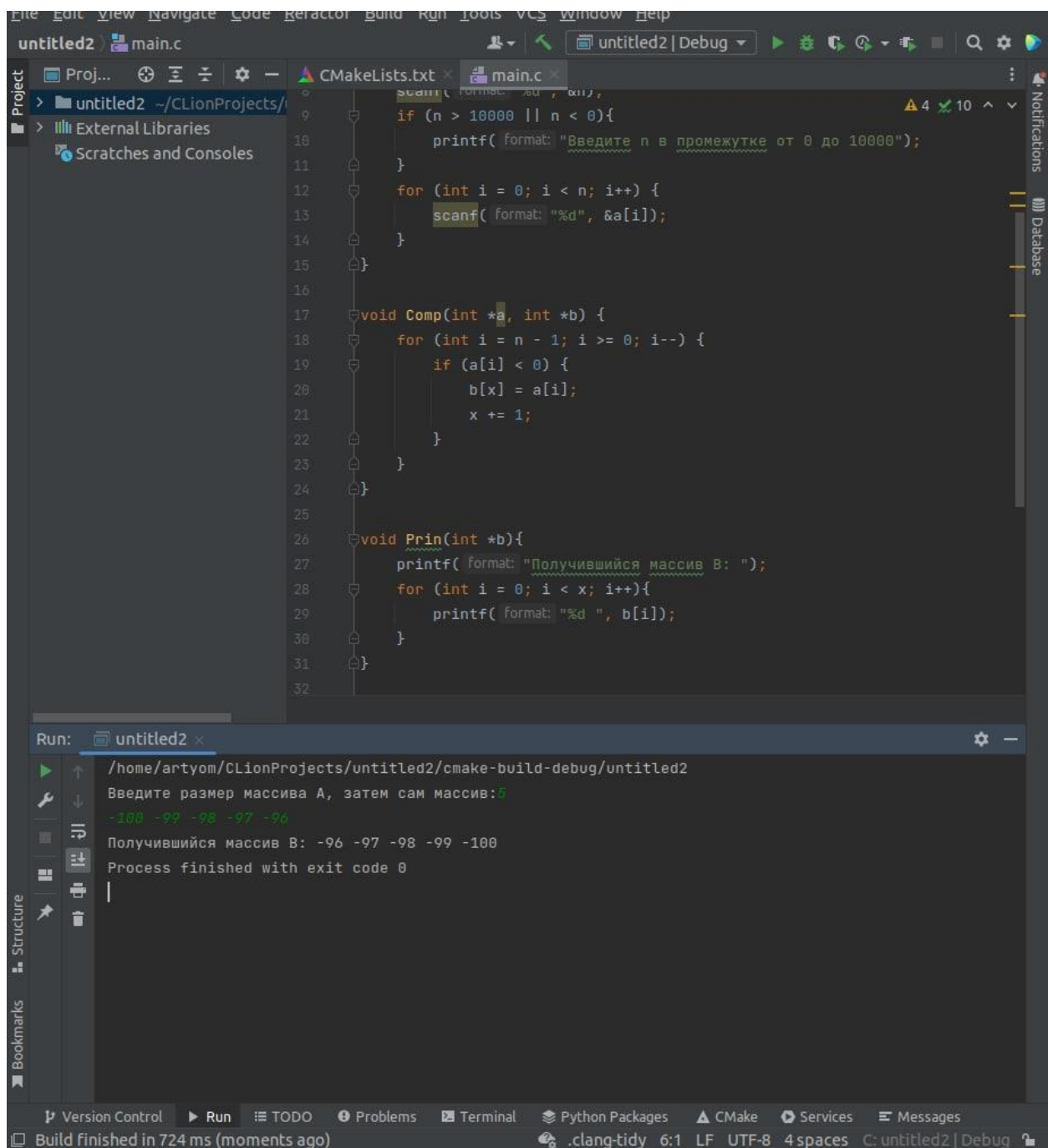
artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./a.out
Введите размер массива А, затем сам массив:-1
Введите n в промежутке от 0 до 10000artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./a.out
Введите размер массива А, затем сам массив:10000000
Введите n в промежутке от 0 до 10000artyom@artyom-GL703VM:~/CLionProjects/untitled2$ ./a.out
Введите размер массива А, затем сам массив:0
Получившийся массив В:
artyom@artyom-GL703VM:~/CLionProjects/untitled2$
```

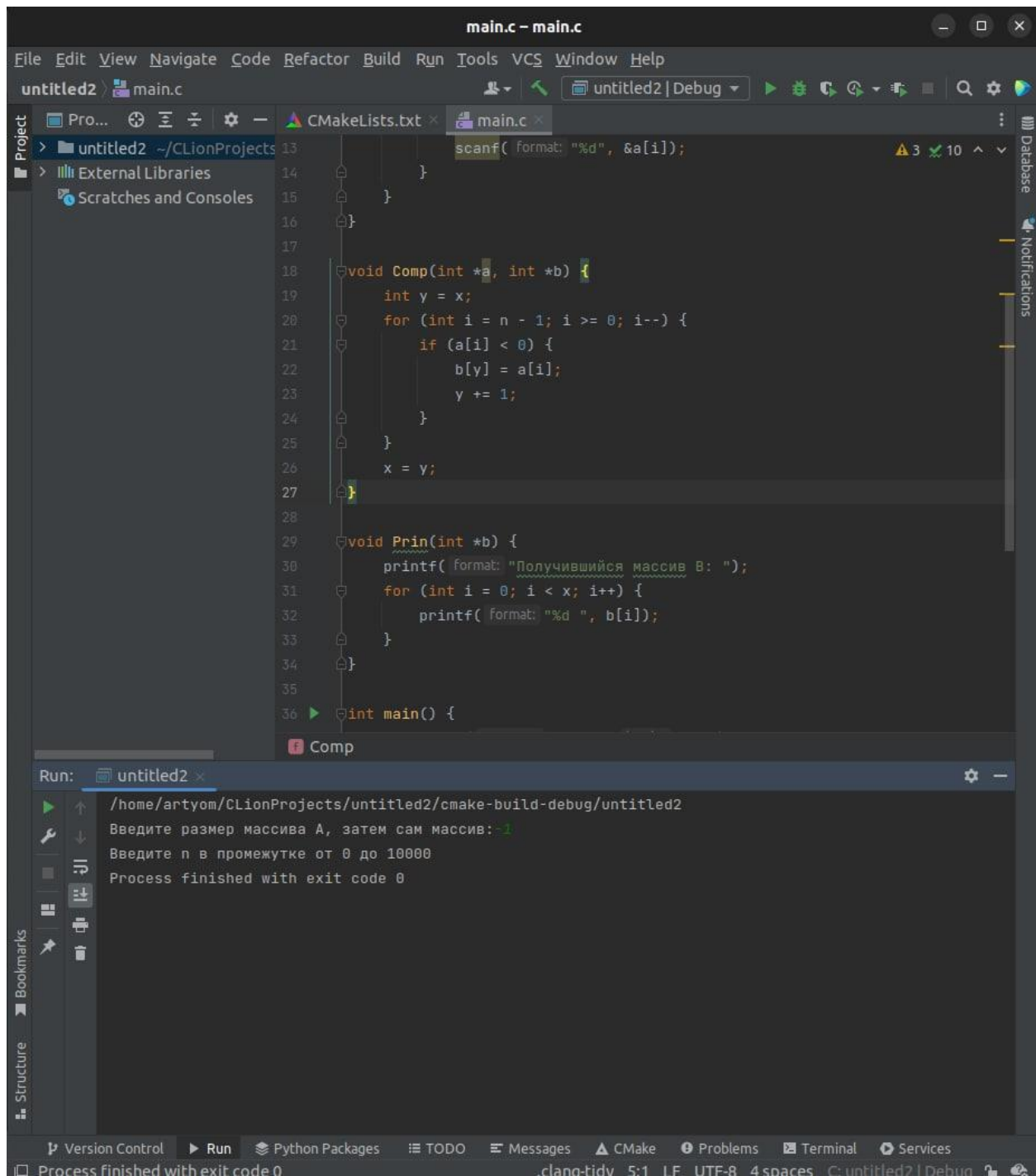


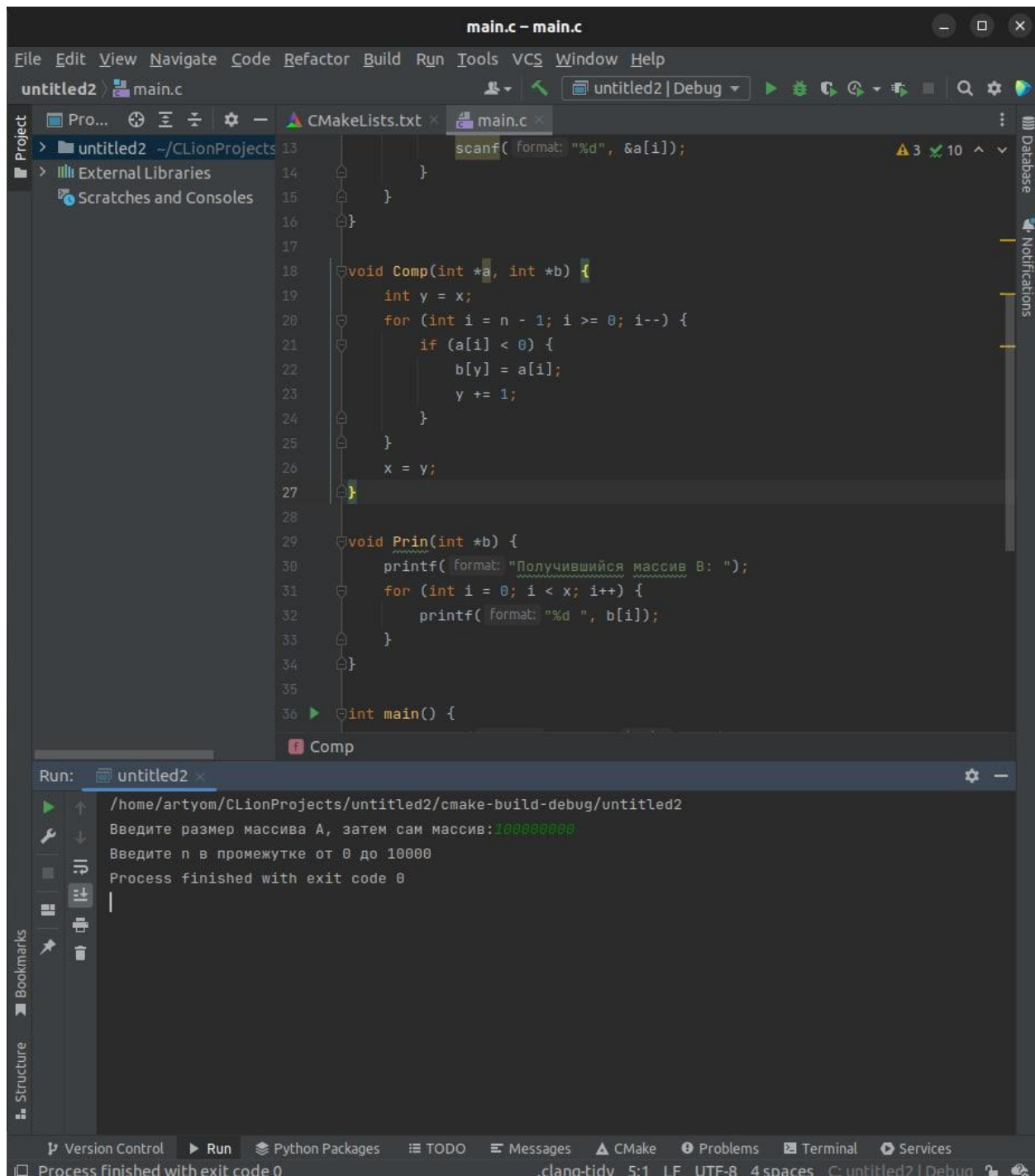


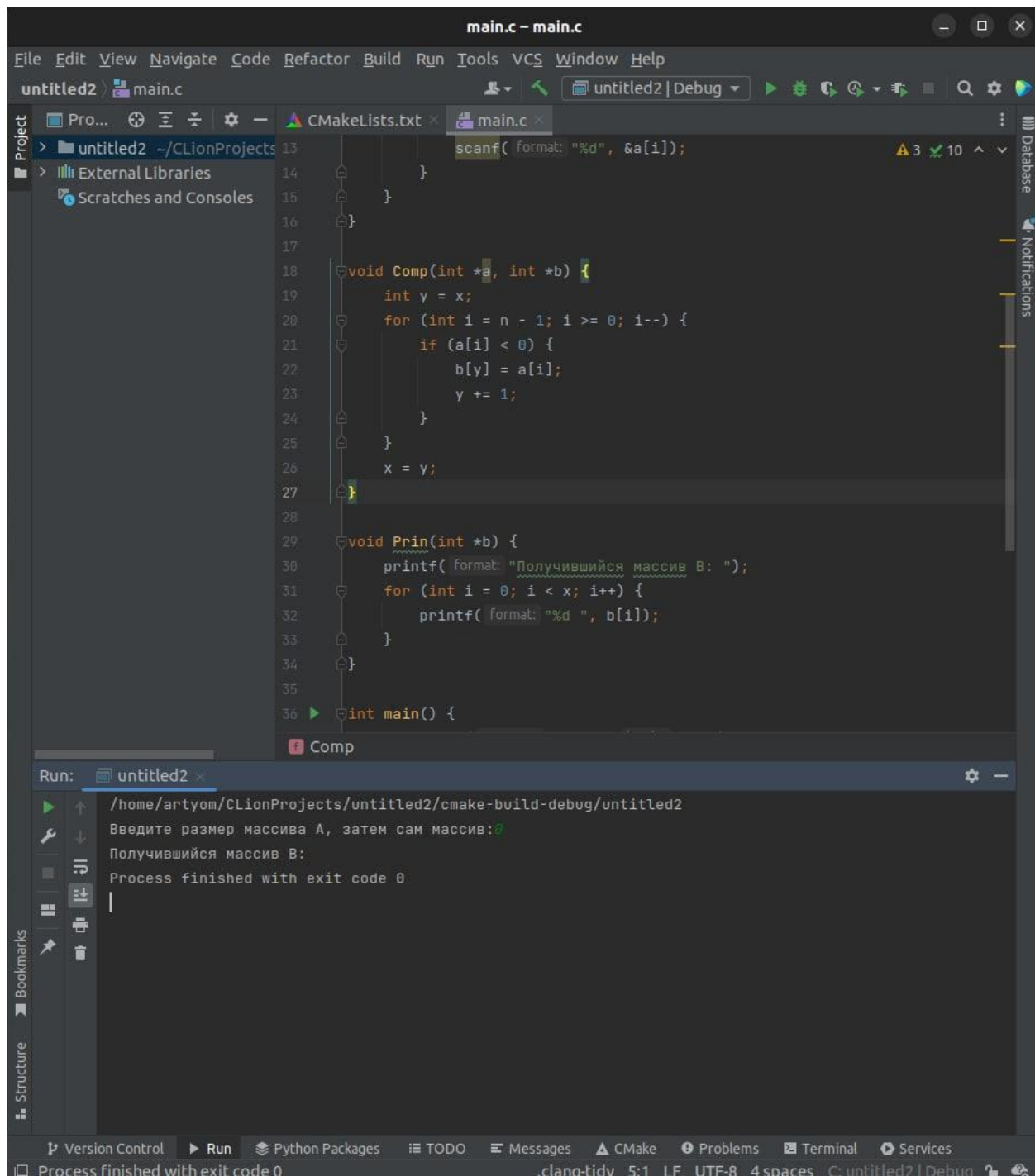












2) Информация, подтверждающая выполнение задания в соответствии требованиям на предполагаемую оценку.

Комментарии:

1)

```
void Scan(int *a) {
    scanf("%d", &n);
    if (n > 10000 || n < 0) {
        printf("Введите n в промежутке от 0 до 10000");
    } else {
        for (int i = 0; i < n; i++) {
            scanf("%d", &a[i]);
        }
    }
}

.text
.globl Scan
.type Scan, @function

Scan:
    # выравнивание
    push    rbp
    mov     rbp, rsp
    sub     rsp, 32

    mov     QWORD PTR -24[rbp], rdi # в rdi адрес a[0]
    lea     rax, n[rip]

    # ввод числа n;
    mov     rsi, rax
    lea     rax, .LC0[rip]
    mov     rdi, rax
    mov     eax, 0
    call    __isoc99_scanf@PLT

    mov     eax, DWORD PTR n[rip]
    # сравниваем n с 10000;
    cmp     eax, 10000
    jg      .L2
    mov     eax, DWORD PTR n[rip]
    # сравниваем с 0
    test    eax, eax
    jns     .L3

.L2:
    # Выводим надпись : "Введите n в промежутке от 0 до 10000"
    lea     rax, .LC1[rip]
```

2)

```
Scan(A);  
  
lea    rax, -80016[rbp] # в rax - адрес a[0];  
mov     rdi, rax # в rdi тоже  
# Вызываем функцию Scan;  
call    Scan  
mov     eax, DWORD PTR n[rip]
```

3)

```
1 void Comp(int *a, int *b) {  
2     int y = x;  
3     for (int i = n - 1; i >= 0; i--) {  
4         if (a[i] < 0) {  
5             b[y] = a[i];  
6             y += 1;  
7         }  
8     }  
9     x = y;  
10 }  
  
87     .globl Comp  
88     .type  Comp, @function  
89 Comp:  
90     push  rbp  
91     mov   rbp, rsp  
92     mov   QWORD PTR -24[rbp], rdi # в rdi адрес a[0]  
93     mov   QWORD PTR -32[rbp], rsi # в rsi адрес b[0]  
94     mov   eax, DWORD PTR x[rip]  
95     mov   DWORD PTR -8[rbp], eax  
96     mov   eax, DWORD PTR n[rip]  
97     sub   eax, 1  
98     mov   DWORD PTR -4[rbp], eax  
99     jmp   .L8  
00 .L10:
```

4)

```
if (n >= 0 && n <= 10000) {  
    Comp(A, B);
```

```
# сравниваем n с 0;  
test    eax, eax  
js      .L15  
mov     eax, DWORD PTR n[rip]  
# сравниваем n с 10000;  
cmp     eax, 10000  
jg      .L15  
lea     rdx, -40016[rbp] # в rdx - адрес b[0];  
lea     rax, -80016[rbp] # в rax - адрес a[0];  
mov     rsi, rdx # в rsi - адрес b[0];  
mov     rdi, rax # в rdi - адрес a[0];  
# Вызываем функцию Comp;  
call    Comp
```

5)

```
void Prin(int *b) {
    printf("Получившийся массив B: ");
    for (int i = 0; i < x; i++) {
        printf("%d ", b[i]);
    }
}

        .globl Prin
        .type  Prin, @function
Prin:
        push   rbp
        mov    rbp, rsp
        sub    rsp, 32
        mov    QWORD PTR -24[rbp], rdi # в rdi адрес b[0]
        lea    rax, .LC2[rip]

        # Выводим текст;
        mov    rdi, rax
        mov    eax, 0
        call   printf@PLT
        mov    DWORD PTR -4[rbp], 0
        jmp    .L12
.L13:
        mov    eax, DWORD PTR -4[rbp]
```

6)

```
        Prin(B);

        lea    rax, -40016[rbp] # в rax - адрес b[0];
        mov    rdi, rax # в rdi - адрес b[0];
        # Вызываем функцию Prin;
        call   Prin
```

Также добавлены другие комментарии, которые не связаны с передачей значений в функции, сюда их включать не стал, но в файле main.s они есть.

Была добавлена локальная переменная y.

```
void Comp(int *a, int *b) {  
    int y = x;  
    for (int i = n - 1; i >= 0; i--) {  
        if (a[i] < 0) {  
            b[y] = a[i];  
            y += 1;  
        }  
    }  
    x = y;  
}
```

```
mov     edx, DWORD PTR -8[rbp]  
movsx   rdx, edx  
lea     rcx, 0[0+rdx*4]  
mov     rdx, QWORD PTR -32[rbp]  
add     rdx, rcx  
mov     eax, DWORD PTR [rax]  
mov     DWORD PTR [rdx], eax  
# добавляем к y 1 (y += 1);  
add     DWORD PTR -8[rbp], 1  
.L9:  
sub     DWORD PTR -4[rbp], 1  
.L8:  
cmp     DWORD PTR -4[rbp], 0  
jns     .L10  
  
# x = y;  
mov     eax, DWORD PTR -8[rbp]  
mov     DWORD PTR x[rip], eax  
nop  
pop     rbp  
ret  
.  
-  
-
```

Были использованы функции Scan(A); Comp(A, B); Prin(B); с передачей данных через параметры (все скрины выше)