ЗАДАНИЕ №3

№1

Для первой программы я добавил точку останова для 28 строки, чтобы отслеживать значение переменных n и result. Оказалось, что из-за того, что у нас постфиксный инкремент, то во время последней итерации цикла, значение переменной result умножается на ноль. Следовательно наш факториал всегда будет нулевым

Для второй программы я добавил точку останова на 40 строчку и обнаружил, что тело цикла ни разу не выполняется. Добавив точку останова на 25 строчку я обнаружил, что во введенном мной массиве на позициях 0, 2, 3, 4 стоят мусорные значения. Добавив точку останова на 52 строчку я обнаружил, что функция по поиску максимума ищет минимум

Для третьей программы я добавил точку останова на 14 строчку и обнаружил следующее:

```
Program received signal SIGFPE, Arithmetic exception.
0x0000555555551e4 in div (a=10, b=0) at task_03.c:21
21          return a / b;
```

Вопросы:

1. Чтобы запросить генерацию отладочной информации следует использовать ключ -g. Если скомпилировать программу без этого ключа, то gdb напишет нам: No debugging symbols found in <имя исполняемого файла>
2. Чтобы запустить программу под отладчиком следует использовать команду: gdb ./<имя исполняемого файла>. Следует использовать команду exit для досрочного завершения
3. list будет продолжать печатать листинг программы с того места, где остановился предыдущий вывод листинга

4. Для вывода значения переменной или выражения следует использовать команду print или ее синоним inspect. Для изменения значения переменной можно использовать команду print <имя переменной>=<значение>

5. step и next

step — пошаговое продолжение выполнения программы с входом вовнутрь вызываемых функций

next — пошаговое выполнение программы без входа в функции

6. backtrace — выводит весь путь к текущей точке останова, то есть названия всех функций, начиная с main()

7. break <имя файла>:<номер строки>

8. Временна точка останова удаляется сразу, как только достигнута

9. disable/enable — выключить или включить точку останова

ignore <n> <col> - col раз пропускает n-ую точку останова

10. condition <номер> <выражение> - задайте <выражение> как условие для точки останова с номером <номер>

11. Точка останова останавливает программу всякий раз, когда ее выполнение достигает определенной точки. Точка наблюдения — специальная точка останова, которая останавливает программу при изменении значения выражения

12. При проверке математических вычислений

13. Для исследования памяти следует использовать команду x

№2

Табличка размеров типов для Linux Ubuntu

char	int	unsigned	long long	short	int32_t	int64_t
1	4	4	8	2	4	8

Размер типов на 10 Windows такой же

№3

Для char:

```
(gdb) x p
0x7fffffffdfc7:	0x958e000a
(gdb) print c=129
$1 = -127 '\201'
(gdb) x p
0x7fffffffdfc7:	0x958e0081
(gdb) print c=-127
$2 = -127 '\201'
(gdb) x p
0x7fffffffdfc7:	0x958e0081
```

Для int:

```
(gdb) x p
0x7fffffffdfc4:	0x0000000a
(gdb) print c=-12930102
$1 = -12930102
(gdb) x p
0x7fffffffdfc4:	0xff3ab3ca
(gdb) print c=1234567891011
$2 = 1912277059
(gdb) x p
0x7fffffffdfc4:	0x71fb0843
```

Для unsigned:

```
(gdb) x p
0x7fffffffdfc4:	0x0000000a
(gdb) print c=-1
$1 = 4294967295
```

(gdb) x p

0x7fffffffdfc4:	0xffffffff

Для long long:

(gdb) x/a p

0x7fffffffdfc0:	0xa

(gdb) print a=6000000000

$1 = 6000000000

(gdb) x/a p

0x7fffffffdfc0:	0x165a0bc00

(gdb) print a=16000000000

$2 = 16000000000

(gdb) x/a p

0x7fffffffdfc0:	0x3b9aca000

№4

(gdb) next

5	int a[10] = { 1, 2, 3, 4, 5 };

(gdb) x a

0x7fffffffdfa0:	0x00000002

(gdb) x a+1

0x7fffffffdfa4:	0x00000000

(gdb) x a+2

0x7fffffffdfa8:	0xbfebfbff

(gdb) x a+3

0x7fffffffdfac:	0x00000000

(gdb) x a+4

0x7fffffffdfb0:	0xffffe3d9

(gdb) x a+5

0x7fffffffdfb4:	0x00007fff

```
(gdb) x a+6
0x7fffffffdfb8:	0x00000064
(gdb) x a+7
0x7fffffffdfbc:	0x00000000
(gdb) x a+8
0x7fffffffdfc0:	0x00001000
(gdb) x a+9
0x7fffffffdfc4:	0x00000000
(gdb) x a+10
0x7fffffffdfc8:	0xb987ef00
(gdb) x a+11
0x7fffffffdfcc:	0x339d32a4
```

№5

```
#include <stdio.h>

int main(void)
{
  int x;
  scanf("%d", x);
  for (int i = 1; i < 1000; ++i)
  {
    if (x % i != 0)
      x *= i;
    else
      x /= i;
  }
}
```

В данном примере удобно использовать точку наблюдения для наблюдения изменения значения выражения x

№6

gdb	QT Creator
break	В конкретной строке на которой вы хотите остановить программу -- щёлкните на поле слева или нажмите **F9** (**F8** для Mac OS X). На функции, в которой вы хотите прерывать программу -- введите имя функции в **Установить точку останова на функцию...** в меню **Отладка**.
run	Чтобы запустить программу под отладчиком выберите в меню **Отладка** пункт **Начать отладку**, или просто нажмите **F5**
delete	Если вы желаете удалить точку останова, просто щёлкните на ней правой кнопкой мыши и выберите **Удалить точку останова** из контекстного меню.
list	Чтобы просмотреть содержимое строки `line`, взгляните на вид **Локальные или наблюдаемые переменные**.