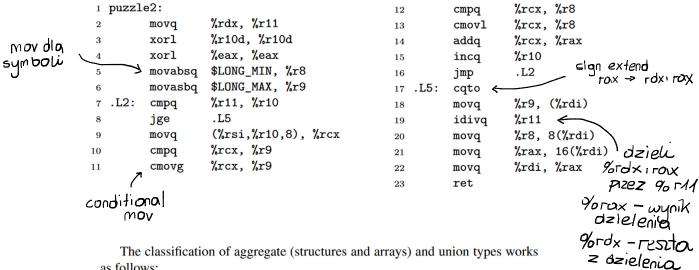
Zadanie 2. Poniżej zamieszczono kod procedury o sygnaturze «struct T puzzle2(long \*a, long n)». Na jego podstawie podaj definicję typu «struct T». Przetłumacz tę procedurę na język C, po czym jednym zdaniem powiedz co ona robi. Wyjaśnij działanie instrukcji «cqto» i «idiv». Gdyby sygnatura procedury nie była wcześniej znana to jaką należałoby wywnioskować z poniższego kodu? Zauważ, że wynik procedury nie mieści się w rejestrach %rax i %rdx, zatem zostanie umieszczony w pamięci. Wskaż regułę w [1, 3.2.3], która wymusza takie zachowanie kompilatora.



as follows:

1. If the size of an object is larger than eight eightbytes, or it contains unaligned fields, it has class MEMORY 12.

2. If the type has class MEMORY, then the caller provides space for the return value and passes the address of this storage in %rdi as if it were the first argument to the function. In effect, this address becomes a "hidden" first argument. This storage must not overlap any data visible to the callee through other names than this argument.

On return %rax will contain the address that has been passed in by the caller in %rdi.

```
12 struct T puzzle2(long *a, long n) {
13
        T result;
        long minimum = LONG_MAX; // %r9
14
15
        long maximum = LONG_MIN; // %r8
        long sum = 0; // %rax
16
        for (int i = 0; i < n; i++) {
17 -
            long elem = a[i];
18
            if (elem < minimum)</pre>
19
                minimum = elem;
20
            if (elem > maximum)
21
22
                maximum = elem;
23
            sum += elem;
        }
long average = sum / n;
24
25
26
        result.a = minimum;
27
        result.b = maximum;
        result.c = average;
28
29
        return result;
30
```

```
12 struct T {
13 long a, b, c;
14 };
```

## bez sygnatury:

```
11
12 long* puzzle2(long *s, long *a, long n);
```

to co robi jest chyba self explainatory