

Напишите функцию `orDone`, которая направляет данные из канала `in` в возвращаемый канал, пока канал `in` открыт и контекст не отменен

```
package main

import (
    "context"
    "reflect"
)

func orDone(ctx context.Context, in <-chan interface{}) <-chan interface{} {
    // напишите ваш код здесь
}

func main() {
    ch := make(chan interface{})
    go func() {
        for i := 0; i < 3; i++ {
            ch <- i
        }
        close(ch)
    }()

    var res []interface{}
    for v := range orDone(context.Background(), ch) {
        res = append(res, v)
    }
    if !reflect.DeepEqual(res, []interface{}{0, 1, 2}) {
        panic("wrong code")
    }
}
```