

Напишите функции `repeatFn` и `take`.

Функция `repeatFn` бесконечно вызывает функцию `fn` и пишет результат ее работы в возвращаемый канал.

Прекращает работу раньше, если контекст отменен.

Функция `take` читает не более чем `num` из канала `in`, пока `in` открыт, и пишет значение в возвращаемый канал.

Прекращает работу раньше, если контекст отменен.

```
package main

import (
    "context"
    "fmt"
    "math/rand"
)

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

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

func main() {
    ctx, cancel := context.WithCancel(context.Background())
    defer cancel()
    rand := func() interface{} { return rand.Int() }
    var res []interface{}
    for num := range take(ctx, repeatFn(ctx, rand), 3) {
        res = append(res, num)
    }
    if len(res) != 3 {
        panic("wrong code")
    }
}
```