

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Функциональное программирование»

ЛАБОРАТОРНАЯ РАБОТА № 1

Студент: Григорьев Г. Г.

Преподаватель: Лаздин А.В.

1. С помощью функций `concat` и `map` реализуйте генерацию всех возможных перестановок элементов списка. Входной список содержит уникальные элементы.

```
> perms [1,2,3]
[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
> perms []
[[]]
```

2. Реализуйте строгую версию левой свёртки списка с помощью оператора `seq`.

```
> foldl (+) 0 [1..1000000]
500000500000
(0.34 secs, 161,693,488 bytes)
> foldlStrict (+) 0 [1..1000000]
500000500000
(0.16 secs, 88,072,480 bytes)
```

3. За один проход свёртки вычислить и сумму, и произведение элементов списка.

Решение:

```
module Test where
```

```
  picks :: [t] -> [(t, t)]
```

```
  picks [] = []
```

```
  picks (x:xs) = [(x,x)] ++ [(x:ys,y) | (ys,y) <- picks xs] --
  всевозможные расщепления голова:хвост
```

```
  perms :: [t] -> [[t]]
```

```
  perms [] = [[]]
```

```
  perms xs = [(x:zs) | (ys,x) <- picks xs, zs <- perms ys]
```

```
  strict :: (a -> b) -> a -> b
```

```
  strict f x = x `seq` f x
```

```
  foldlppp :: (a -> b -> a) -> a -> [b] -> a
```

```
  foldlppp _ z [] = z
```

```
  foldlppp f z (x:xs) = foldlppp f (f z x) xs
```

```
  foldl' :: (a -> b -> a) -> a -> [b] -> a
```

```
foldl' _ z [] = z
```

```
foldl' f z (x:xs) = strict (foldl' f) (f z x) xs
```

```
sumProduct [] = (0,0)
```

```
sumProduct xs = (sum,product)
```

```
where (sum,product) =
```

```
  foldr(\x (y,n)->(x+y,n*x)) (0,1) xs
```