

Сборник контрольных работ для подготовки к экзамену

Д.Н. Лавров
2018

Контрольная №1.

Умение программировать

- **Вариант 1**
- Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 6; 2; 9; -3 ; 6 ответ: 4.
- **Вариант 2**
- Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите алгоритм, позволяющий найти и вывести количество пар элементов массива, сумма которых нечётна и положительна. Под парой подразумевается два подряд идущих элемента массива.

Решение КР №1

#Решение Вариант 1

```
a = []
n = 20
for i in range(0, n):
    a.append(int(input()))
k = 0
for i in range(0, n-1):
    if (a[i]%3==0 or a[i+1]%3==0):
        k+=1
print(k)
```

#Решение Вариант 2

```
N = 20
i = None
j = None
a = [int(input()) for i in range(N)]
k = 0
for i in range(N-1):
    if (a[i]+a[i+1])%2 !=0 and a[i]+a[i+1]>0:
        k += 1
print(k)
```

Контрольная №2.

Проектирование цикла с помощью инварианта

- **Вариант 1**

- Дана последовательность из 20 элементов. Найти среднее значение всех элементов делящихся на 7. Гарантируется, что хотя бы одно такое значение в последовательности есть.

1. Определить что будет множеством M
2. Определить предикат $P(x)$, где $x \in M$
3. Определить преобразование $T(x)$
4. Сформулировать инвариант $I(x)$
5. Сформулировать условие окончания $Q(x)$
6. Показать, что $I(x) \text{ and } Q(x) \Rightarrow P(x)$
7. Показать, что $I(x_0) == \text{True}$
8. Показать сохранение инварианта при преобразовании $T(x)$. То есть
$$I(x) == \text{True} \Rightarrow I(T(x)) == \text{True}$$
9. Написать и протестировать программу:
 $x = x_0$
while not $Q(x)$:
 $x = T(x)$

- **Вариант 2**

- Дана последовательность из 20 элементов. Найти среднее значение всех элементов при делении на 8, дающих в остатке 5. Гарантируется, что хотя бы одно такое значение в последовательности есть.

КР №2. Решение варианта 1.

1. $x=(s,c,k)$, x in M
2. $P(x)$: «В s содержится сумма всех элементов, делящихся на 7»
3. $T(x)$:
 $T(sk+1,ck+1,k)= \begin{cases} (s_k+y_k, c_k+1, k+1), & \text{если } y_k \% 7 == 0 \\ (s_k, c_k, k+1), & \text{иначе} \end{cases}$
4. $I(x)$: В s/c содержится среднее всех элементов до k -ого, делящихся на 7.
5. $Q(x)$: $k \geq 20$
6. Очевидно, что если $k \geq 0$, то s/c содержится среднее всех элементов до 20-ого, делящихся на 7.
7. $X_0=(0,0,0)$. $s/c==0/0$ — неопределенное значение, но первое же значение, где $y \% 7 == 0$ даёт $I(x)=True!!!$
8. По условию задачи обязательно найдётся хотя бы один элемент, делящийся на 7. Как только он встретится, то $I(x) == True$ для любого следующего k .

Продолжение

9. Решение задач

Первый вариант решения

```
s=0; c=0; k=0
```

```
while not k>=20:
```

```
    y=m[k]  % это другое m – не M
```

```
    if y%7==0:
```

```
        s+=y; c+=1
```

```
    k+=1
```

```
print(s/c)
```

- Второй вариант решения

```
s=0; c=0
```

```
for y in m:
```

```
    if y%7==0:
```

```
        s+=y; c+=1
```

```
print(s/c)
```

Контрольная №3. Построение индуктивных функций (30 мин.)

- Напишите программу, печатающую значение многочлена степени $n > 0$ в заданной точке x_0 . Коэффициенты многочлена хранятся в массиве w в порядке убывания степеней и являются целыми числами, также как и значение x_0 . Величины n , x_0 и элементы массива A изменять в программе нельзя. Функцию или оператор вычисления степени не использовать. Разработать однопроходный алгоритм.
1. Опишите индуктивную функцию на последовательности коэффициентов. Покажите как $F(w*x)$ выражается через $F(w)$ и x . Запишите это правило в виде функции $G(F(w),x)$, так что $F(w*x)=G(F(w),x)$
 2. Определите значение F на пустой последовательности. Обозначим это значение F_0 .
 3. Напишите код на Python
- ```
F=F0
for w in A:
 F=G(F,x)
print(F)
```

# Кр№3. Возможные решения

```
def G(fw, x):
 return fw*x0+x
```

```
n=int(input())
x0=int(input())
f0=0
w=[]
for i in range(n+1):
 w.append(int(input()))
```

```
for x in w:
 f0=G(f0, x)
print(f0)
```

```
n=int(input())
x0=int(input())
f0=0
for i in range(n+1):
 x=int(input())
 f0=f0*x0+x

print(f0)
```



# Контрольная №4 FILTER

Перепишите код  
в функциональном стиле (модуль functools)

## • Вариант №1

```
res=[]
lst = [2,3,4,5,6]
for x in lst:
 if (x%2==0) & (x>3) :
 res.append(x)
print(res)
```

## • Вариант №2

```
res=[]
x = [2,3,4,5,6]
for i in range(len(x)) :
 if
 (x[i]%3==0) & (x[i]>2) :
 res.append(x[i])
print(res)
```

# ОТВЕТЫ КР№4

#Вариант №1

```
lst = [2, 3, 4, 5, 6]
```

```
res=list(filter(lambda x: (x%2==0) & (x>3), lst))
```

```
print(res)
```

#Вариант №2

```
x = [2, 3, 4, 5, 6]
```

```
res=list(filter(lambda x: (x%3==0) & (x>2), x))
```

```
print(res)
```

# Контрольная №5 MAP-REDUCE

Переписать код в функциональном стиле используя map и reduce

- **Вариант 1**

```
lst= [1, 3, 5, 7, 9]
```

```
s=0
```

```
for x in lst:
```

```
 s+=2*x+5
```

```
print(s)
```

- **Вариант 2**

```
x= [1, 3, 5, 7, 9]
```

```
s=0
```

```
for i in range(len(lst)):
```

```
 s+=x[i]**2-2*x[i]
```

```
print(s)
```

# ОТВЕТЫ КР№5

- Вариант1
- `s= reduce(lambda x, r: r+x, map(lambda x: 2*x+5,lst))`
- `print (s)`
  
- Вариант 2
- `s= reduce(lambda x, r: r+x, map(lambda x:x**2-2*x,x))`
- `print (s)`

# ОТВЕТЫ КР№5

- Вариант1
- `s= reduce(lambda r, x: r+2*x+5,lst,0)`
- `print (s)`
  
- Вариант 2
- `s= reduce(lambda r, x: r+x**2-2*x,x,0)`
- `print (s)`

# Контрольная работа №6

## REGXP

- Дана строка:
- `inp=["192.168.1.123", "192.168.12.123", "44", "-5", "1.3", \`  
`"-4.39991", "23.32.1", "252.321.321.1", \`  
`"299.299.299.299", "0.0.0.0"]`
- **Задание 1.** Напишите регулярное выражение и код, его использующий, для распечатывания **целых чисел с плавающей точкой** (разделитель точка, целые попадают в этот формат).
- 
- **Задание 2.** IPv4-адрес представляет собой 32-битное значение в десятично-точечной нотации 4 октета (в каждом по 8 бит). Напишите регулярное выражение для нахождения IPv4-адреса, и найдите все IP-адреса в указанной выше строке.
- Примеры верных и неверных IPv4 адресов.
- 192.168.1.123 – Верный
- 1.16.133.133 – Верный
- 252.321.321.1 – Некорректный
- .... – Совсем не IP
- 299.299.299.299 – Некорректный
- Время выполнения - 30 мин. Можно пользоваться тетрадями с лекциями.

# Возможное решение КР№6

*«У программиста была  
проблема, которую он  
начал решать регэкспами.  
Теперь у него две  
проблемы»*

Народная программистская  
мудрость

- *Задание 1. Традиционное решение:*

```
r=re.compile("^\\-?\\d+\\.\\d*\\)?$")
for s in inp:
 if r.match(s): print(s)
```

- *Задание 2. С использованием «функциональности»:*

```
rs="^(25[0-5]|2[0-4]\\d|[0-1]?\\d?\\d?\\.){3}" + \
 "(25[0-5]|2[0-4]\\d|[0-1]?\\d?\\d?)$"
r=re.compile(rs)
lst=filter(r.match,inp)
print(list(lst))
```