|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Информатика и системы управления

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Программное обеспечение ЭВМ и информационные технологии

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Савинов Егор Дмитриевич

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

ИУ7-14Б

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Учебная

Тип практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Название предприятия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Савинов Е.Д.

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2018 г.*

Оглавление

[Условие задачи 3](#_Toc532856511)

[Схема программы (по ГОСТ) 4](#_Toc532856512)

[Описание программы 9](#_Toc532856513)

[Текст программы с комментариями 9](#_Toc532856514)

[Заключение 11](#_Toc532856515)

[Список литературы 11](#_Toc532856516)

Условие задачи

Римская рулетка

Исторически известный Флавиус Джозефус рассказывает, как в Римско-Еврейском конфликте в 67 г. до н.э. римляне взяли город Джотапата, которым он командовал. Убегая, Джозефус попал в западню в пещере вместе с 40 соратниками. Римляне обнаружили его местоположение и предложили ему сдаться, но его соратники отказались это сделать. Он предложил, что они убьют друг друга, способом, который выбрало большинство. Традиционно, действуя для всех, было стать в круг, и, начиная с какой-либо точки, каждого третьего убивать. Единственный оставшийся в живых был Джозефус, сдавшийся римлянам. Спрашивается: тренировался ли заранее Джозефус в тишине с 41 камнем в темном месте или он математически вычислил, что он должен занять 31 позицию для спасения?

Прочитав отчет об этом событии, вы будете одержимы идеей попасть в схожую ситуацию в ближайшем будущем. Для подготовки себя к такому событию вы решили написать программу, чтобы заставить ваш ПК вычислить позицию, с которой должен начинаться счет, чтобы вы остались в живых. На практике, ваша программа должна быть в состоянии просмотреть следующие варианты процесса, описанного Джозефусом. n > 0 людей становятся в круг, лицом друг к другу и нумеруются от 1 до n по часовой стрелке. Ваш номер 1. Начиная с позиции i, счет начинается по часовой стрелке, и так пока мы не дойдем до человека k (k > 0), которого убивают. Затем мы считаем дальше k человек, начиная с человека, который только что покинул этот мир. Считая от человека, который выбыл, до человека k, который будет убит, и так далее, пока не останется один. К примеру, когда n = 5, k = 2, I = 1, порядок ввода 2, 5, 3 и 1. Оставшийся в живых 4.

Ввод и вывод.

Ваша программа должна считывать строки, содержащие значения n и k (в этом порядке), и для каждой строки выдавать номер человека, с которого нужно начать счет, чтобы остаться в живых. К примеру, в предшествующем случае начальная позиция 3. Ввод обрывается строкой, содержащей нули для n и k.

Максимум 100 человек могут принять участие.

Пример ввода.

1 1

1 5

0 0

Пример вывода.

1

1

Начало

Схема программы (по ГОСТ)

text = [[‘1’,’1’],

[‘1’,’5’],

[‘0’,’0’]]

Цикл k

От 0 до len(text)

j != 1

Цикл i

От 1 до n + 1

win\_number = 0

l = 3

n = int(text[k][0])

k = int(text[k][1])

p = i

j = n

people = [‘0’] \* n

num1 = 0

fl1 = 0

fl3 = 0

Нет

Да

fl1 == 1

Да

Нет

Нет

Да

Num1 != k

Нет

Да

fl1 == 0 и people[p-1] == ‘0’

Да

Нет

p == n

p = p + 1 – n

p = p + 1

fl1 = 1

people[p-1] ==’0’

Нет

Да

num1 = num + 1

num1 == k

Да

Нет

p = p + 1

fl3 = 0

p == n

Нет

Да

p = p – n + 1

fl3 = 1

Нет

p != n и fl3 == 0

Да

fl3 = 0

num1 = 0

people[p-1] = ‘1’

j = j – 1

num = 0

num == 1

Увеличить v на 1

Цикл v

num = num + 1

Цикл v

От 0 до len(people)

Нет

people[v] == ‘0’

Да

Нет

Да

Цикл v

От 0 до len(people)

Нет

people[v] == ‘0’

Да

Нет

people[l - 1] != ‘1’

Да

win\_number = i

fl1 = 1

Увеличить v на 1

Цикл v

Увеличить i на 1

Цикл i

Print(‘Если игра начнется с места’, win\_number, ‘вы выживите на месте’,l,’.’)

Нет

Да

win\_number != 0

Да

Нет

n == 1

print(‘1’)

Увеличить k на 1

Цикл k

конец

Описание программы

Программа вычисляет позицию, с которой должен начинаться счет, чтобы человек под номером l остался в живых. Алгоритм моделирует ситуацию, когда n (n > 0) человек становятся лицом друг к другу и нумеруются от 1 до n по часовой стрелке. Игра начинается с человека под номером p. Когда доходит до k-го (k > 0) человека, его убивают. Затем считают дальше k человек, начиная с человека, который только что умер и так далее, пока не останется один выживший. Алгоритм выводит выигрышную стратегию для человека под номером l, то есть номер места (win\_number), с которого должна начаться игра, чтобы человек под номером l, выжил.

Текст программы с комментариями

text = [['1','1'],

['1','5'],

['0','0']]

for k in range(len(text)):

n = int(text[k][0])

k = int(text[k][1])

l = 3

win\_number = 0

for i in range(1, n + 1):

p = i

j = n

people = ['0'] \* n

num1 = 0 # счет k

fl1 = 0 # флаг для корректной работы в цикле

fl3 = 0 # флаг для корректной работы цикла по

# кругу

if fl1 == 1:

break

while j != 1: # цикл, пока не останется один человек в

# в живых

while num1 != k: # цикл отсчитывает k людей

if fl1 == 0 and people[p-1] == '0': # один раз не включает первого

# человека в счет, с которого

#начиналась игра

if p == n:

p = 1 + p - n

else:

p += 1

fl1 = 1

if people[p-1] == '0': # счет k

num1 += 1

if num1 == k: # остановиться на k-ом человеке

break

if p == n: # циклит хождение по массиву

p = 1 + p - n

fl3 = 1

if p != n and fl3 == 0: # переходит на следующего человека

p += 1

fl3 = 0

fl3 = 0

num1 = 0

people[p-1] = '1' # убивают человека

j -= 1

num = 0

for v in range(len(people)): # проверка на последнего выжившего

if people[v] == '0':

num += 1

if num == 1:

for v in range(len(people)):

if people[v] == '0':

if people[l - 1] != '1': # если человек на месте l не умер, то

# задача решена

win\_number = i

fl1 = 1

break

if win\_number != 0:

print('\nЕсли игра начнется с места', win\_number, 'вы выживите на месте', l, '.')

if n == 1:

print('1')

Заключение

Учебно-вычислительный практикум укрепил мои знания в области алгоритмов.

Я научился распознавать их в задачах реального мира и взял себе в привычку, анализировать эффективность кода, который я пишу.

Список литературы

# Data Structure Practice: for Collegiate Programming Contests and Education

[электронный ресурс]. – Режим доступа: <https://books.google.ru/books?id=E22mCwAAQBAJ&pg=PA163&lpg=PA163&dq=roman+roulette+in+particular+your+program+should+be+able&source=bl&ots=CRoL32SAeU&sig=4p5BJ7YxkcnUQlWjtlubs_0nlIo&hl=en&sa=X&ved=2ahUKEwjfxsO6sp3fAhWxpIsKHf0YAUsQ6AEwA3oECAcQAQ#v=onepage&q&f=false> (дата обращения 16.12.2018)