**# Задача №1. Создайте программу для игры с конфетами человек против бота.**

**# Условие задачи: На столе лежит 2021 конфета. Играют два игрока делая ход друг после друга.**

**# Первый ход делает человек. За один ход можно забрать не более чем 28 конфет.**

**# Все конфеты оппонента достаются сделавшему последний ход.**

**# Сколько конфет нужно взять первому игроку, чтобы забрать все конфеты у своего конкурента?**

**# 2021 21 ---> 2000 бот4 -> 1996 .... бот --->29 --> 27 >> 2конф**

print("-" \* 70)

print('Здравствуйте! Вас приветствует игра "Забери все конфеты!" ')

print('Основные правила игры: за один ход мы можем взять не более 28 конфет, ')

print('Итак, начнём!')

print("-" \* 70)

from random import randint

candy=int(2021)

max\_candy = int(28)

bot\_move = max\_candy

for i in range(candy,0,-1):

if i % 2 == 1:

print("Человек берет конфет: ", end='')

move = int(input())

if candy > max\_candy:

if move <= max\_candy:

candy = candy-move

print('Осталось -', candy)

else:

print('Недопустимое количество взятых конфет, максимально можно взять 28, попробуйте еще раз! ')

move=int(input())

if move <= max\_candy:

candy = candy-move

print('Осталось -', candy)

else:

if move <= candy:

candy = candy-move

print('Осталось -', candy)

if candy==0:

print('Вы победили!')

break

else:

print('Недопустимое количество взятых конфет, Вы хотите взять больше, чем осталось, попробуйте еще раз! ')

print('Осталось -', candy)

move=int(input())

if move==candy:

print('Вы победили!')

break

if i % 2 == 0:

print("Бот берет конфет: ", end='')

if candy < max\_candy:

bot\_move=candy

move2 = randint(1,bot\_move)

print(move2)

candy = candy-move2

print('Осталось -', candy)

if candy == 0:

print('Победил БОТ!')

break

**# Задача №1.2. Добавьте игру против бота**

print("-" \* 70)

print('Здравствуйте! Вас приветствует игра "Забери все конфеты!" ')

print('Основные правила игры: за один ход мы можем взять не более 28 конфет, ')

print('Итак, начнём!')

print("-" \* 70)

from random import randint

candy=int(2021)

max\_candy = int(28)

bot\_move = max\_candy

for i in range(candy,0,-1):

if i % 2 == 1:

print("Бот 1 берет конфет: ", end='')

if candy < max\_candy:

bot\_move=candy

move2 = randint(1,bot\_move)

print(move2)

candy = candy-move2

print('Осталось -', candy)

if candy == 0:

print('Победил 1 БОТ!')

break

if i % 2 == 0:

print("2 Бот берет конфет: ", end='')

if candy < max\_candy:

bot\_move=candy

move2 = randint(1,bot\_move)

print(move2)

candy = candy-move2

print('Осталось -', candy)

if candy == 0:

print('Победил 2 БОТ!')

break

**# Задача №2. Создайте программу для игры в ""Крестики-нолики"".**

print("\*" \* 22)

print(" Игра Крестики-нолики ")

print( "\*" \* 22)

board = list(range(1,10))

def draw\_board(board): ***# создание поля для игры***

print("-" \* 13)

for i in range(3):

print("|", board[0+i\*3], "|", board[1+i\*3], "|", board[2+i\*3], "|")

print("-" \* 13)

def input\_step(cross\_or\_zero): ***# функция для ввода Х или 0***

valid = False

while not valid:

player\_answer = input("Ход игрока: " + cross\_or\_zero +"? ")

try:

player\_answer = int(player\_answer)

except:

print("Некорректный ввод. Вы уверены, что ввели число?")

continue

if player\_answer >= 1 and player\_answer <= 9: ***# проверяем введен.число попадает в разреш.диапозон***

if(str(board[player\_answer-1]) not in "XO"): ***# вторая проверка, запрашиваемая клетка пустая?***

board[player\_answer-1] = cross\_or\_zero ***# если да, вносим Х или 0***

valid = True

else:

print("Клетка занята! Попробуйте еще раз!")

else:

print("Некорректный ввод. Возможное число от 1 до 9")

def check\_win(board): ***# функция на проверку выигрышной позиции***

win\_position = ((0,1,2), (3,4,5), (6,7,8), (0,3,6), (1,4,7), (2,5,8), (0,4,8), (2,4,6)) for step in win\_position:

if board[step[0]] == board[step[1]] == board[step[2]]: ***# проверяем возможные комбинации***

return board[step[0]]

return False

def basic(board): ***# основная функция игры:***

counter = 0

win = False

while not win:

draw\_board(board) ***# вызов функции - поле игры***

if counter % 2 == 0:

input\_step("X") ***# если ход четный, вводим Х***

else:

input\_step("O") ***# если ход четный, вводим 0***

counter += 1

if counter > 4: ***# если ходов сделано более 4 вызываем функции на проверку*** выигрышной позиции

tmp = check\_win(board)

if tmp:

print(tmp, "выиграл!")

win = True

break

if counter == 9: ***# если ходов сделано 9 и неопределена победа - ничья***

print("Ничья!")

break

draw\_board(board) ***# вызов функции - поле игры***

basic(board) ***# вызов основ***

**# Задача №3: Реализуйте RLE алгоритм: реализуйте модуль сжатия и восстановления данных.**

**# stroka = "aaabbbbccbbb"**

**# ....**

**# stroka = "3a4b2c3b"**

**# Входные данные хранятся в отдельных текстовых файлах.**

**text.txt: aaabbbbccbbb**

print("-" \* 22)

print('Алгоритм сжатия:')

print("-" \* 22)

**text2.txt: 3a4b2c3b**

with open("text.txt", "r") as f:

text = f.read()

print('Первоначальная строка: ', text)

code = '' ***# создаем пустую строку, для сохранения результата***

i = 0

while i < len(text):

count = 1 ***# подсчитываем кол-во вхождений символа в индексе `i`***

while i + 1 < len(text) and text[i] == text[i + 1]: ***# условие: если след.индекс меньше размера строки и если***

***элемент равен следующему элементу***

count = count + 1 ***# увеличиваем счетчик***

i = i + 1 ***# переходим к следующему индексу***

code += str(count) + text[i]  ***# добавляем в итоговую строку количество вхождений и***

***текущий символ***

i = i + 1

print('Cтрока после сжатия: ', code)

print("-" \* 22)

print("-" \* 22)

print('Алгоритм восстановление данных:')

print("-" \* 22)

with open("text2.txt", "r") as f:

text2 = f.read()

print('Первоначальная строка: ', text2)

count = ''

decode = '' ***# создаем пустую строку, для сохранения результата***

for i in text2:

if i.isdigit(): ***# если элемент списка является числом, то записываем его в count***

count += i

else:

decode += i \* int(count) ***# если нет, то умножаем переменную списка на count***

count = ''

print('Cтрока после восстановления: ', decode)

print("-" \* 22)