

Задания к работе

Задача 1. Последовательность Фибоначчи.

Описание программы: Числа Фибоначчи – это ряд чисел, в котором каждое следующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13, Иногда ряд начинают с нуля: 0, 1, 1, 2, 3, 5, В данном случае мы будем придерживаться первого варианта. Пользователю необходимо ввести произвольное число, далее, программа выводит ряд чисел.

ЗАДАНИЕ: Отладьте код программы, зафиксируйте работу программы скриншотами в отчете и потом напишите новую функцию fibRecurse, которая использует рекурсивный алгоритм вычисления чисел Фибоначчи, который основан на рекуррентных отношениях чисел Фибоначчи:

$$F_1 = F_2 = 1, F_n = F_{n-1} + F_{n-2}.$$

Код программы:

```
def fibSequence(n):
    assert n > 0
    series = [1]
    while len(series) < n:
        if len(series) == 1:
            series.append(1)
        else:
            series.append(series[-1] + series[-2])
    for i in range(len(series)):
        series[i] = str(series[i])
    return(', '.join(series))

print(fibSequence(int(input('Сколько чисел? '))))
```

Задача 2. Простые числа

Описание программы: Простое число — это число, у которого только два делителя: 1 и само число. Программа запрашивает у пользователя разрешение на вывод следующего просто числа.

ЗАДАНИЕ: Отладьте код программы, зафиксируйте работу программы скриншотами в отчете и измените программу так, чтобы она запрашивала у пользователя число и отвечала – простое оно или нет, и, если число не простое, то выводила на экран БЛИЖАЙШЕЕ простое число.

Код программы:

```
def isPrime(x):
    if x == 2:
        return True
    if x % 2 == 0:
        return False
```

```

        for i in range(3, int(x**0.5)+1, 2):
            if x % i == 0:
                return False
        return True

def genPrime(currentPrime):
    newPrime = currentPrime + 1
    while True:
        if not isPrime(newPrime):
            newPrime += 1
        else:
            break
    return newPrime
currentPrime = 2

while True:
    answer = input('Показать следующее простое число? (Y/N) ')
    if answer.lower().startswith('y'):
        print(currentPrime)
        currentPrime = genPrime(currentPrime)
    else:
        break

```

Задача 3 Калькулятор

Описание программы: программа выполняет над двумя вещественными числами одну из четырех арифметических операций (сложение, вычитание, умножение или деление).

ЗАДАНИЕ: Отладьте код программы, зафиксируйте работу программы скриншотами в отчете и добавьте в калькулятор операции: возведение в степень, вычисление процента числа, извлечение корня.

Код программы:

```

def calc(a, b, op):
    if op not in '+-/*':
        return 'Пожалуйста, выберите тип операции: "+, -, *, /"'
    if op == '+':
        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a + b))
    if op == '-':
        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a - b))
    if op == '*':
        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a * b))
    if op == '/':
        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a / b))
def main():
    a = int(input('Пожалуйста, введите первое число: '))
    b = int(input('Пожалуйста, введите второе число: '))

```

```

op = input(
    'Какой вид операции Вы желаете осуществить?\n'
    'Выберите между "+, -, *, /" : ')
print(calc(a, b, op))

if __name__ == '__main__':
    main()

```

Задача 4. Игра «21 point»

Описание программы: Изначально у пользователя 0 очков. Программа спрашивает у пользователя, хочет ли он взять карту. Пользователь должен ввести в консоль либо у – yes, либо n – no. Если пользователь ответил n, то программа озвучивает набранное количество очков и завершает свою работу. Если пользователь ввел команду “y”, то игра продолжается и выдается 1 карта из списка. Далее, программа прибавляет к числу очков снятую карту и выводит количество очков. Если количество очков больше 21, то пользователь проигрывает, и программа завершает свою работу. Если число очков равно 21, то пользователь выиграл. Если меньше - программа снова предлагает взять карту. После окончания игры, программа прощается с пользователем.

ЗАДАНИЕ: Отладьте код программы, зафиксируйте работу программы скриншотами в отчете и добавьте для бота коэффициент – уровень сложности (он же – уровень подглядывания в колоду) – число от 0 до 1. Чем ближе к 1, тем сильнее играет бот. На последнем ходе бота делается проверка случайного числа, и если оно больше коэффициента, то последний ход бот делает случайно, а если меньше – то бот автоматически подбирает нужную карту, чтобы выиграть.

Код программы:

```

import random
import os
import time

# Счет
score_playera = 0
score_bota = 0

# Начальное сообщение
all_carts = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
print("Поиграем в 21? \nЕсли хотите играть нажмите Enter, если хотите выйти, то нажмите Ctrl+C")
input()
# Цикл
while True:
    if score_playera == 21:
        print("Больше карт не надо, у вас 21")
        print("Вы автоматически победили бота, так как у вас 21.")
        input("Нажмите Enter, чтобы закрыть окно."); break
    if score_playera > 21:
        print("Вы проиграли, так как набрали больше 21")
        print("Попробуйте свою попытку в другой раз.")
        input("Нажмите Enter, чтобы закрыть окно."); break
    yes_or_no = input("Будете ли вы брать карту?\nВведите yes, если хотите брать карту или введите no, если не берете карту.\n")
    os.system('cls')
    if yes_or_no == 'yes':
        os.system('cls')

```

```

score_carts = random.choice(all_carts)
print("Вы взяли карту выпало:", score_carts)
score_playera += score_carts
print("Сейчас у вас ", score_playera)
if yes_or_no == 'no':
    print("У вас ", score_playera, "очков.")
    print("Ход бота")
    time.sleep(3)
    os.system('cls')
    while True:
        if score_bota < 15:
            print("Бот берет карту")
            score_carts = random.choice(all_carts)
            print("Боту выпало", score_carts, "очков.")
            score_bota += score_carts
            print("У бота ", score_bota, "очков.")
            time.sleep(3)
            os.system('cls')
        if score_bota > 21:
            print("Бот проиграл.\nТак как у него", score_bota, "очков, а у вас ",
score_playera)
            input("Нажмите Enter, чтобы закрыть"); exit(0)
        if score_bota > score_playera:
            print("Бот победил.\nТак как у него", score_bota, "очков, а у вас ",
score_playera, "\nНе расстраивайтесь. Попробуйте ещё раз.")
            input("Нажмите Enter, чтобы закрыть"); exit(0)
        if score_bota == score_playera:
            print("Вы набрали равное количество очков и у вас ничья")
            input("Нажмите Enter, чтобы закрыть"); exit(0)

```