

Тема: Составление программ циклической структуры.

ПЗ-4.1

Постановка задачи:

Найти произведение всех целых чисел от A до B включительно.

Тип алгоритма:

Циклический

Блок-схема:



Текст программы на Python:

```
def program():
```

```
    try:
```

```
        # запрашиваем ввод чисел
```

```
        A = int(input("Введите число A: "))
```

```
        B = int(input("Введите число B: "))
```

```
        answer = 1
```

```
        while A != B: #запускаем цикл, который работает, пока A не равно B
```

```
            A += 1
```

```
            answer *= A
```

```
        print(answer) #выводим ответ на экран
```

```
    except ValueError: #если ошибка, запускаем программу заново
```

```
        print("ошибка ввода")
```

```
        program()
```

```
program()
```

Протокол работы программы:

```
Введите число А: 1
Введите число В: 3
6
```

ПЗ-4.2

Постановка задачи:

Определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней K (целое) и суммарный пробег S (вещественное число).

Тип алгоритма:

Циклический

Блок-схема:



Текст программы:

```
def program():
    try:
        P = float(input("Введите количество процентов: ")) # запрашиваю количество процентов
        fd = 10 # Первый день
        fd = fd + fd / 100 + fd # узнаю процент пробега за первый день
        if 0 < P < 50:
            tfd = 0
            days = 1
            while tfd <= 200: # цикл выполняется, пока суммарный пробег не превысит 200
                fd += fd / 100 * P
                tfd += fd
                days += 1
            print(f"Количество дней: {days} \nСуммарный пробег: {tfd}") # вывод результата
```

```
else:
```

```
    print("Число должно быть от 1 до 50") # если меньше 1 или больше 50, заново  
    program()
```

```
except ValueError:
```

```
    print("ошибка ввода")  
    program()
```

```
program()
```

Протокол работы программы:

```
Введите количество процентов: 23
```

```
Количество дней: 7
```

```
Суммарный пробег: 264.73237822837893
```