

Отчет по большой лабораторной по курсу «Машинное обучение»

ВЫПОЛНИЛИ СТУДЕНТЫ ГРУППЫ 5040102/10201

ДАНИЛ ПЕСТРЯКОВ

ВАСИЛИСА КУДРЯВЦЕВА

Введение

В данном отчете представлены результаты, который получены в ходе тестирования нашей программы на различных входных данных.

Метрика качества построена следующим образом:

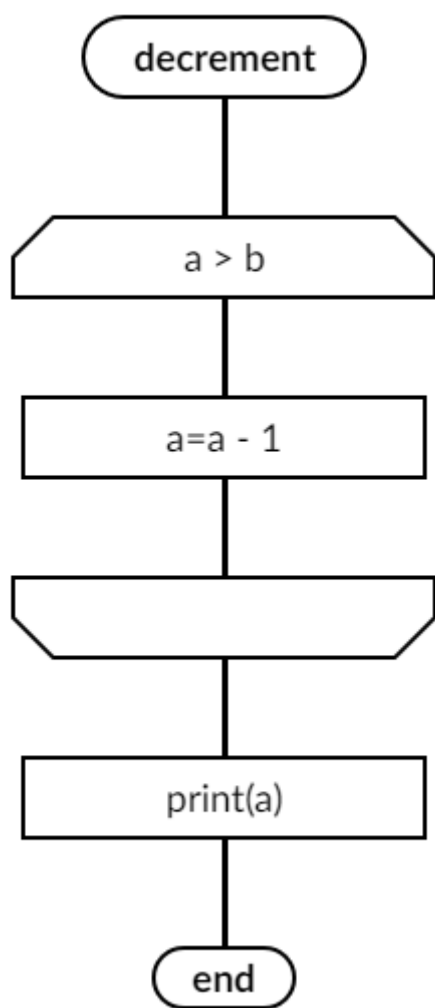
Если количество строк не совпадает, то она равна модулю разницы количества строк, деленного на количество строк в эталоне и умноженное на 0.2. Выводим это число.

Если количество строк совпадает, то ставим 0.2 и для каждой строки проверяем, совпадает ли количество табов. Если не совпадает, то к 0.2 нужно добавить количество строк, в котором совпадает количество табов, умноженное на 0.4. Выводим это число.

Если количество табов во всех соответствующих строках совпадает, то метрика уже 0.6. К ней прибавляется число, рассчитанное следующим образом. Для каждой пары строк считаем расстояние Левенштейна с помощью встроенного модуля python. Берем среднее по всем строкам и прибавляем к 0.6 это среднее, умноженное на 0.4. Отметим, что расстояние Левенштейна выдается нормированным к 1.

Таким образом, получаем объективную метрику, которая принимает значение от 0 до 1 включительно и по которой можно судить, в какой из частей преобразования блоксхемы в код возникли проблемы.

Файл 20221214231758.png:



Результат работы программы:

Run test 20221214231758.png:

Generated code:

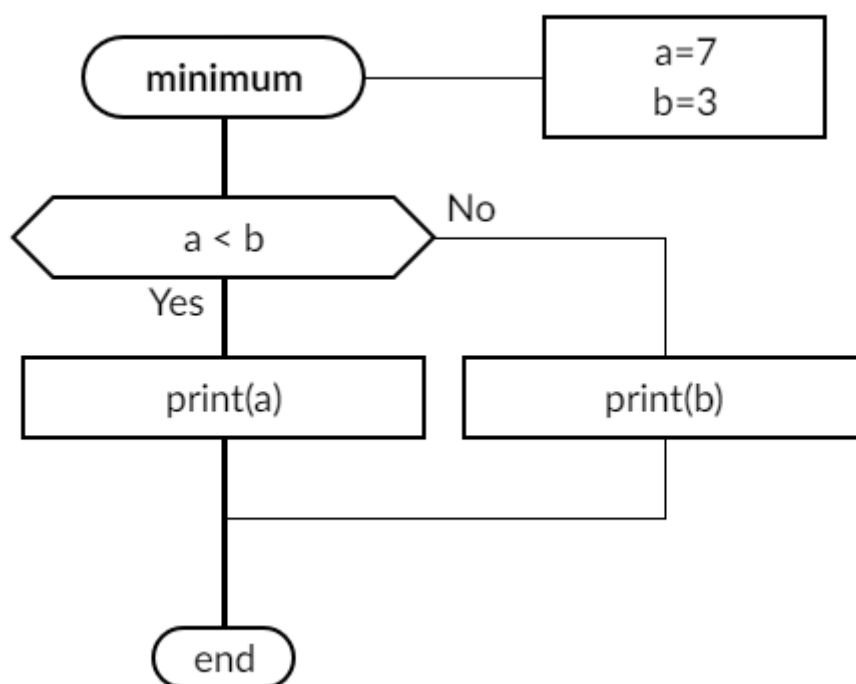
```
def decrement():  
    while a>b:  
        a=a-1  
    print(a)
```

Original code:

```
def decrement():  
    while a>b:  
        a=a-1  
    print(a)
```

Metric: 1.0

Файл 20221216222919.png:



Результат работы программы:

Run test 20221216222919.png:

Generated code:

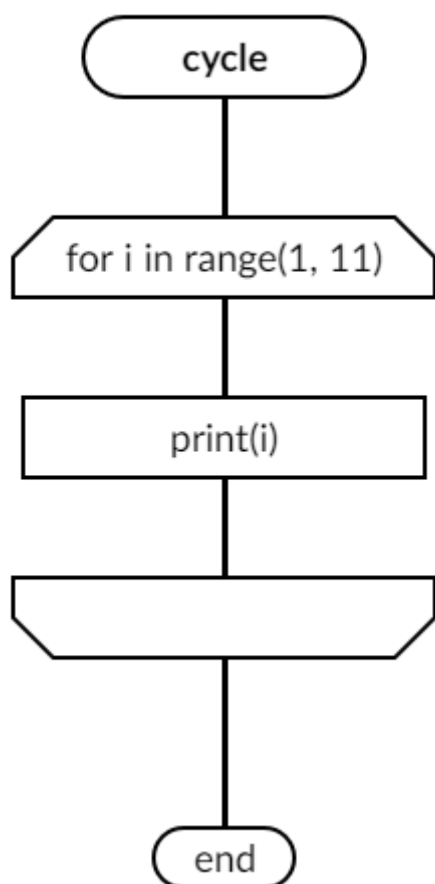
```
def minimum(a=7, b=3):  
    if a<b:  
        print(a)  
    else:  
        print(b)
```

Original code:

```
def minimum(a=7, b=3):  
    if a<b:  
        print(a)  
    else:  
        print(b)
```

Metric: 1.0

Файл 20221216223250.png:



Результат работы программы:

Run test 20221216223250.png:

Generated code:

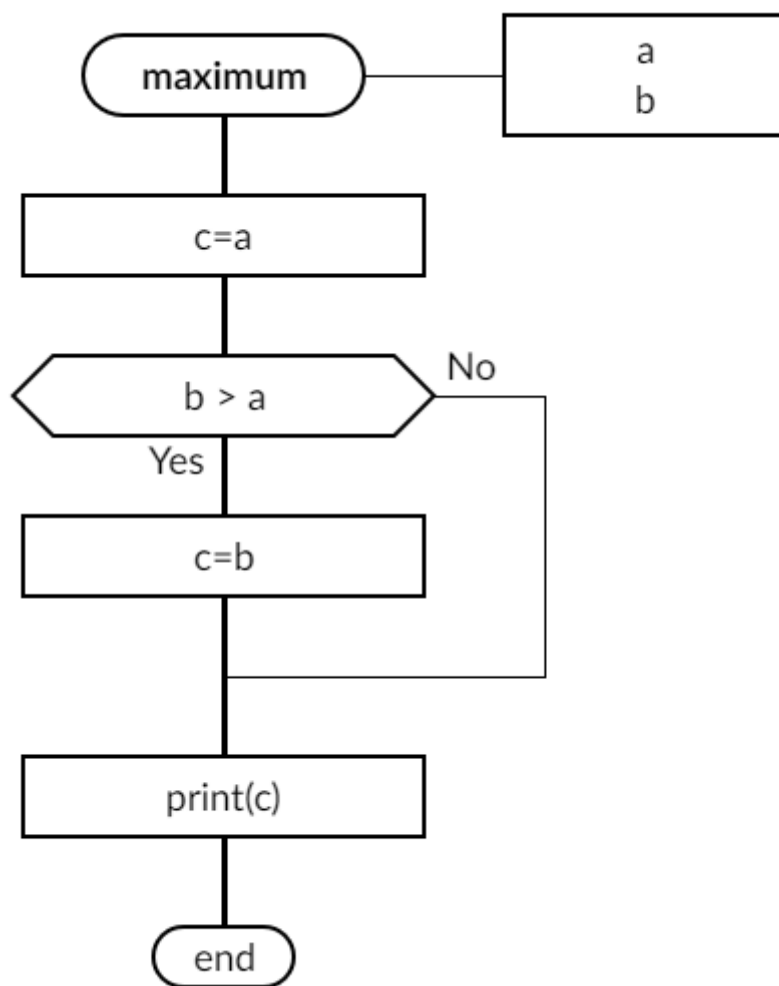
```
def cycle():  
    for i in range(1, 11):  
        print(i)
```

Original code:

```
def cycle():  
    for i in range(1, 11):  
        print(i)
```

Metric: 1.0

Файл 20221217125146.png:



Результат работы программы:

Run test 20221217125146.png:

Generated code:

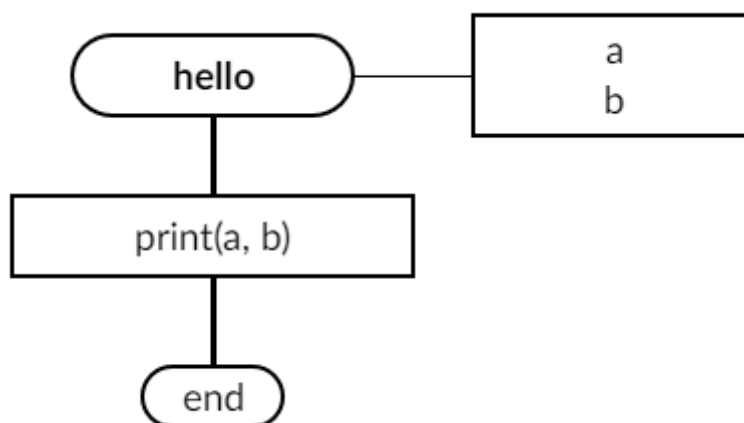
```
def maximum(a, b):  
    c=a  
    if b>a:  
        c=b  
    print(c)
```

Original code:

```
def maximum(a, b):  
    c=a  
    if b>a:  
        c=b  
    print(c)
```

Metric: 1.0

Файл 20221217125240.png:



Результат работы программы:

Run test 20221217125240.png:

Generated code:

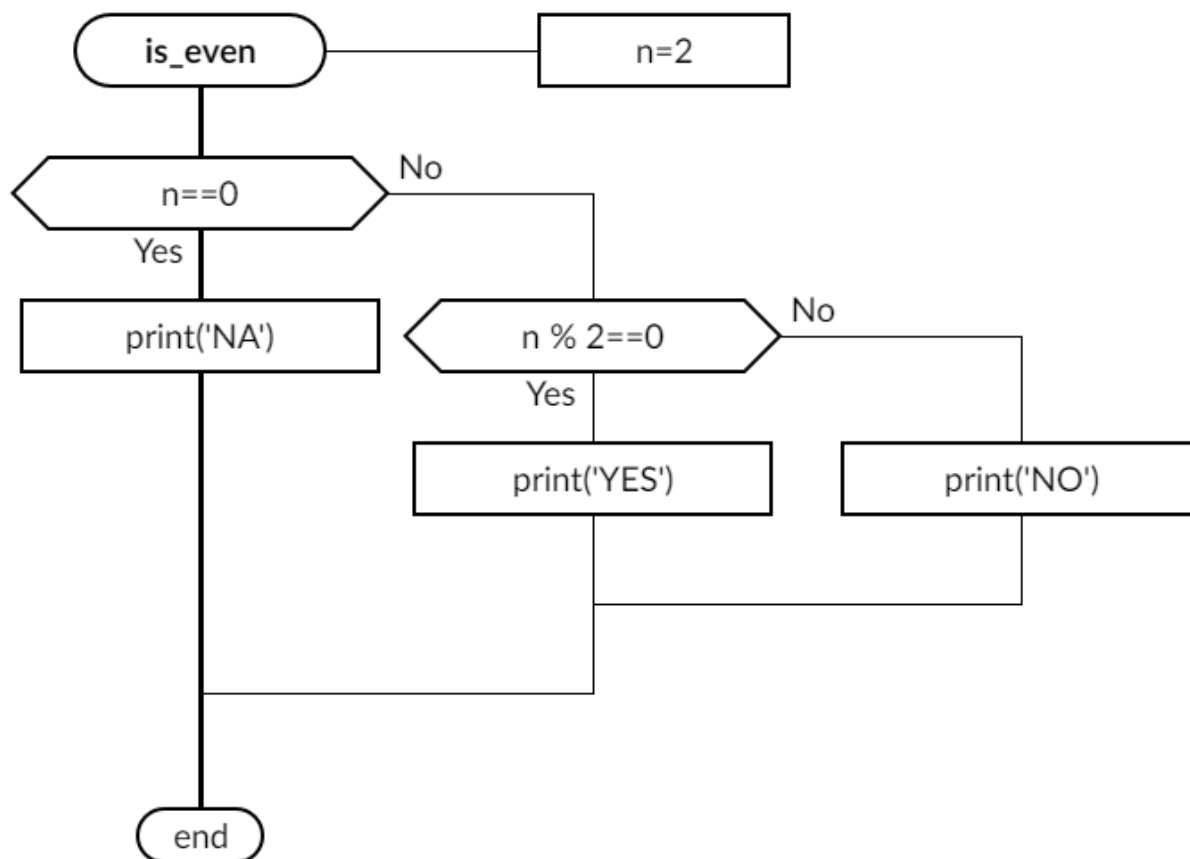
```
def hello(a, b):  
    print(a, b)
```

Original code:

```
def hello(a, b):  
    print(a, b)
```

Metric: 1.0

Файл 20221217210344.png:



Результат работы программы:

Run test 20221217210344.png:

Generated code:

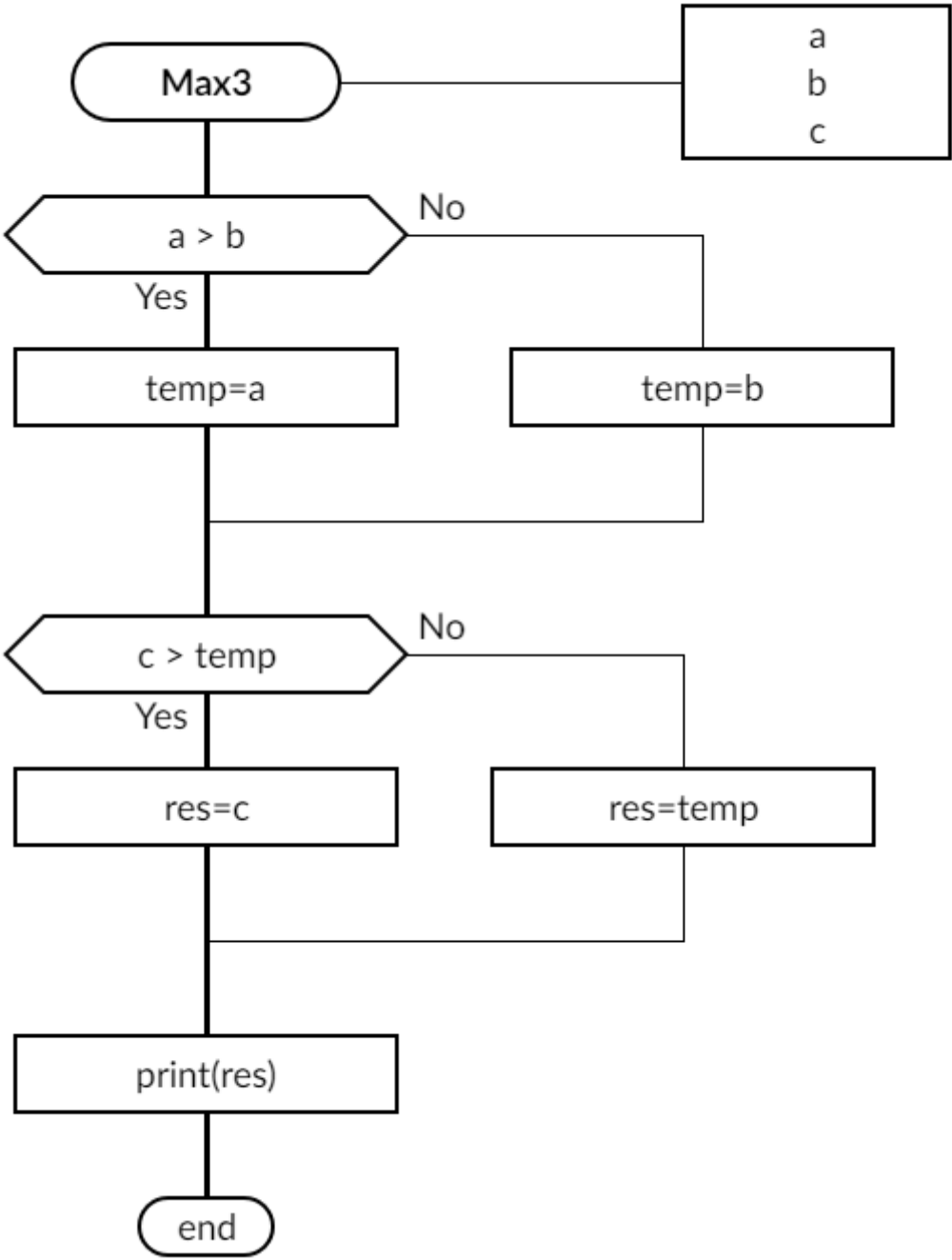
```
def is_even(n=2):  
    if n==0:  
        print('NA')  
    else:  
        if n%2==0:  
            print('YES')  
        else:  
            print('NO')
```

Original code:

```
def is_even(n=2):  
    if n==0:  
        print('NA')  
    else:  
        if n%2==0:  
            print('YES')  
        else:  
            print('NO')
```

Metric: 0.9905000000000002

Файл 20221220211317.png:



Результат работы программы:

Generated code:

```
def Max3(a, b, Cc):  
    if a>b:  
        temp=a  
    else:  
        temp=b  
    if c > temp:  
        res=c  
    else:  
        res=temp  
    print(res)
```

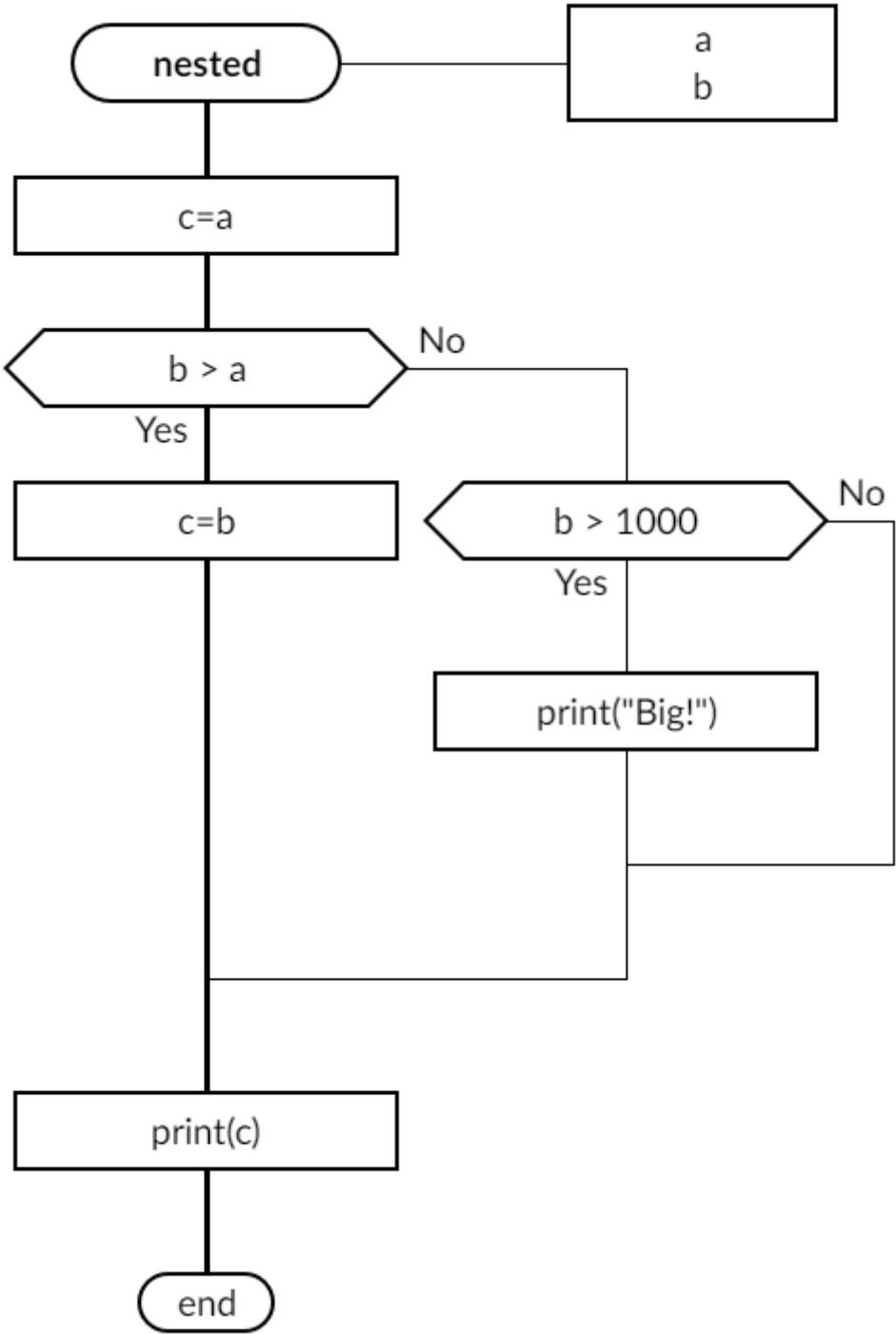
Original code:

```
def Max3(a, b, c):  
    if a>b:  
        temp=a  
    else:  
        temp=b  
    if c>temp:  
        res=c  
    else:  
        res=temp  
    print(res)
```

Metric: 0.9956

Text detected

Файл 20221226122142.png:



Результат работы программы:

Run test 20221226122142.png:

Generated code:

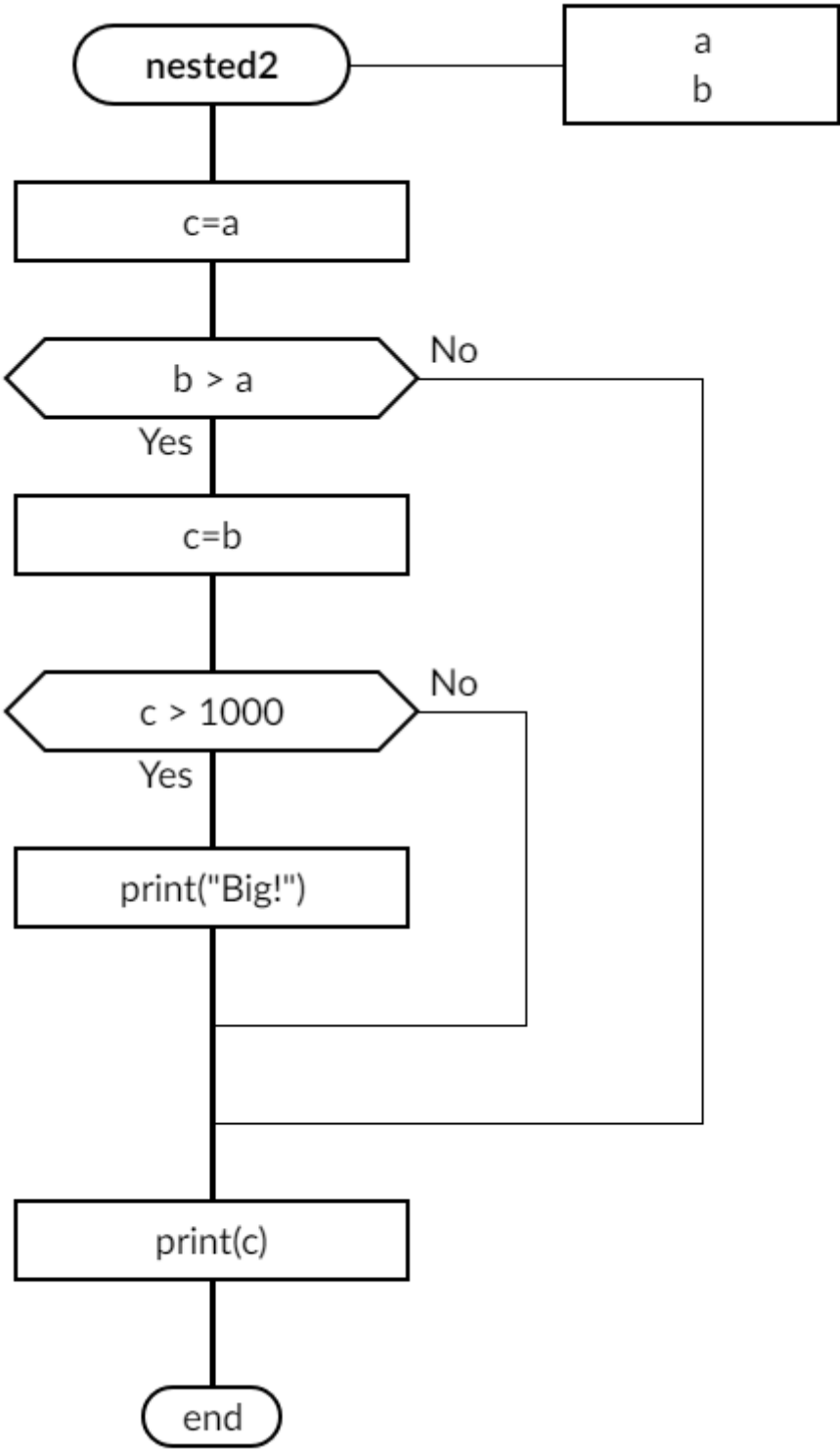
```
def nested(a, b):  
    c=a  
    if b>a:  
        c=b  
    else:  
        if b > 1000:  
            print("Big!")  
    print(c)
```

Original code:

```
def nested(a, b):  
    c=a  
    if b>a:  
        c=b  
    else:  
        if b>1000:  
            print("Big!")  
    print(c)
```

Metric: 0.9960000000000001

Файл 20221226122501.png:



Результат работы программы:

Run test 20221226122501.png:

Generated code:

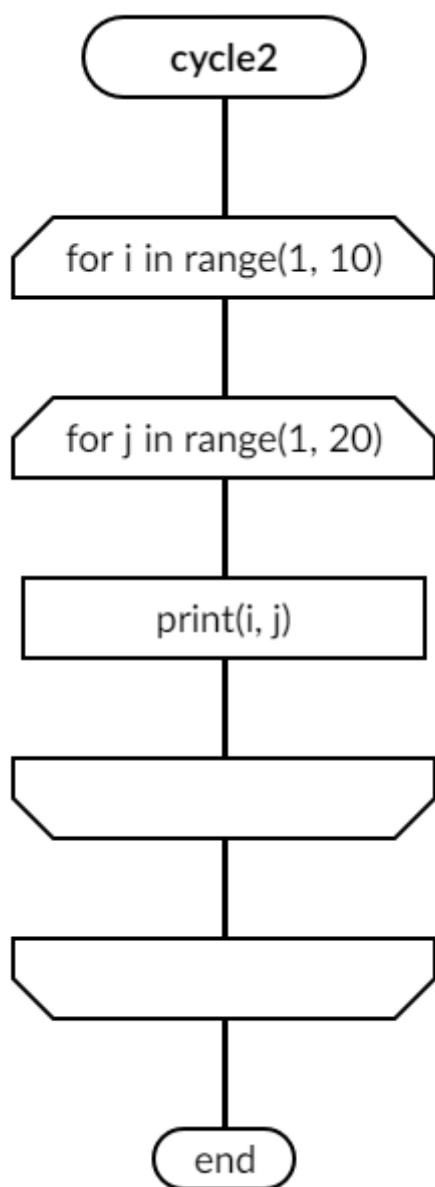
```
def nested2(a, b):  
    c=a  
    if bea:  
        c=b  
        if c > 1000:  
            print("Big!")  
    print(c)
```

Original code:

```
def nested2(a, b):  
    c=a  
    if b>a:  
        c=b  
        if c>1000:  
            print("Big!")  
    print(c)
```

Metric: 0.9885714285714287

Файл 20221226223928.png:



Результат работы программы:

Run test 20221226223928.png:

Generated code:

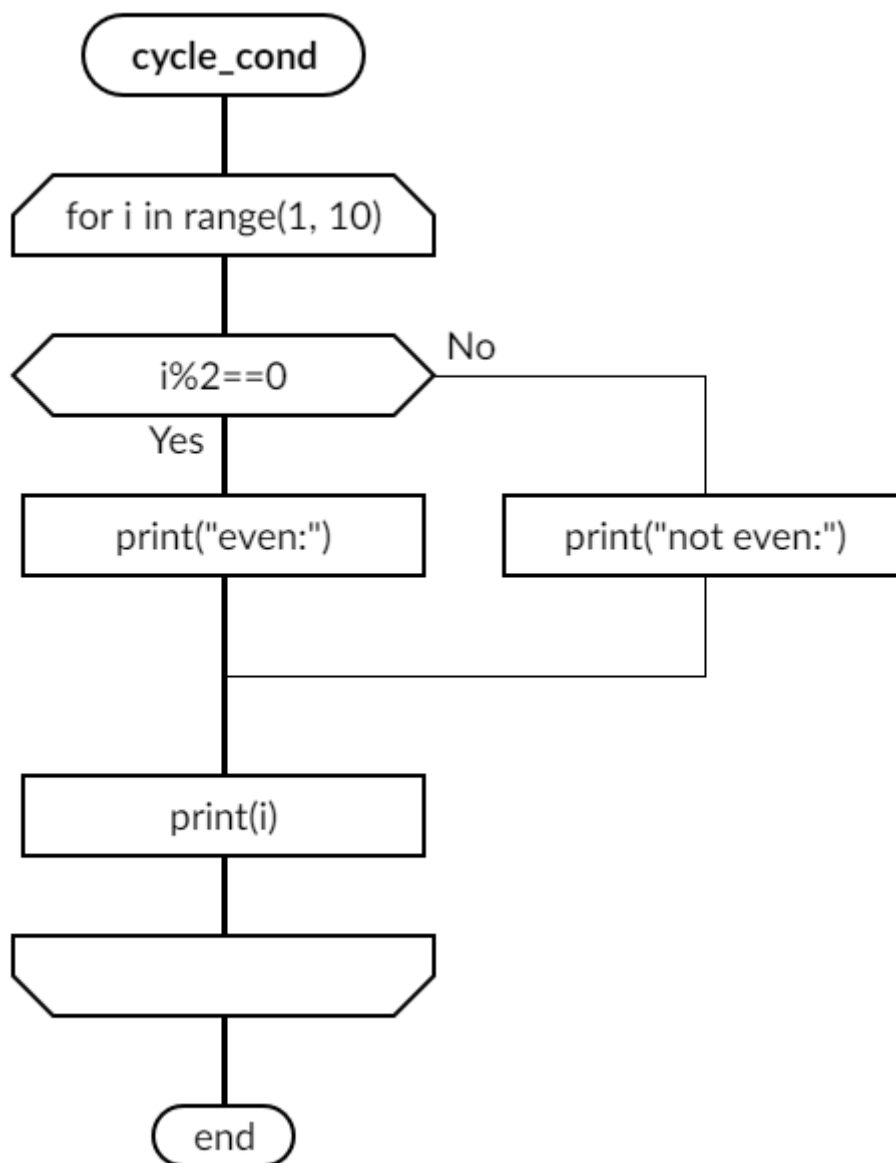
```
def cycle2():  
    for i in range(1, 10):  
        for j in range(1, 20):  
            print(i, j)
```

Original code:

```
def cycle2():  
    for i in range(1, 10):  
        for j in range(1, 20):  
            print(i, j)
```

Metric: 1.0

Файл 20221226235839.png:



Результат работы программы:

Run test 20221226235839.png:

Generated code:

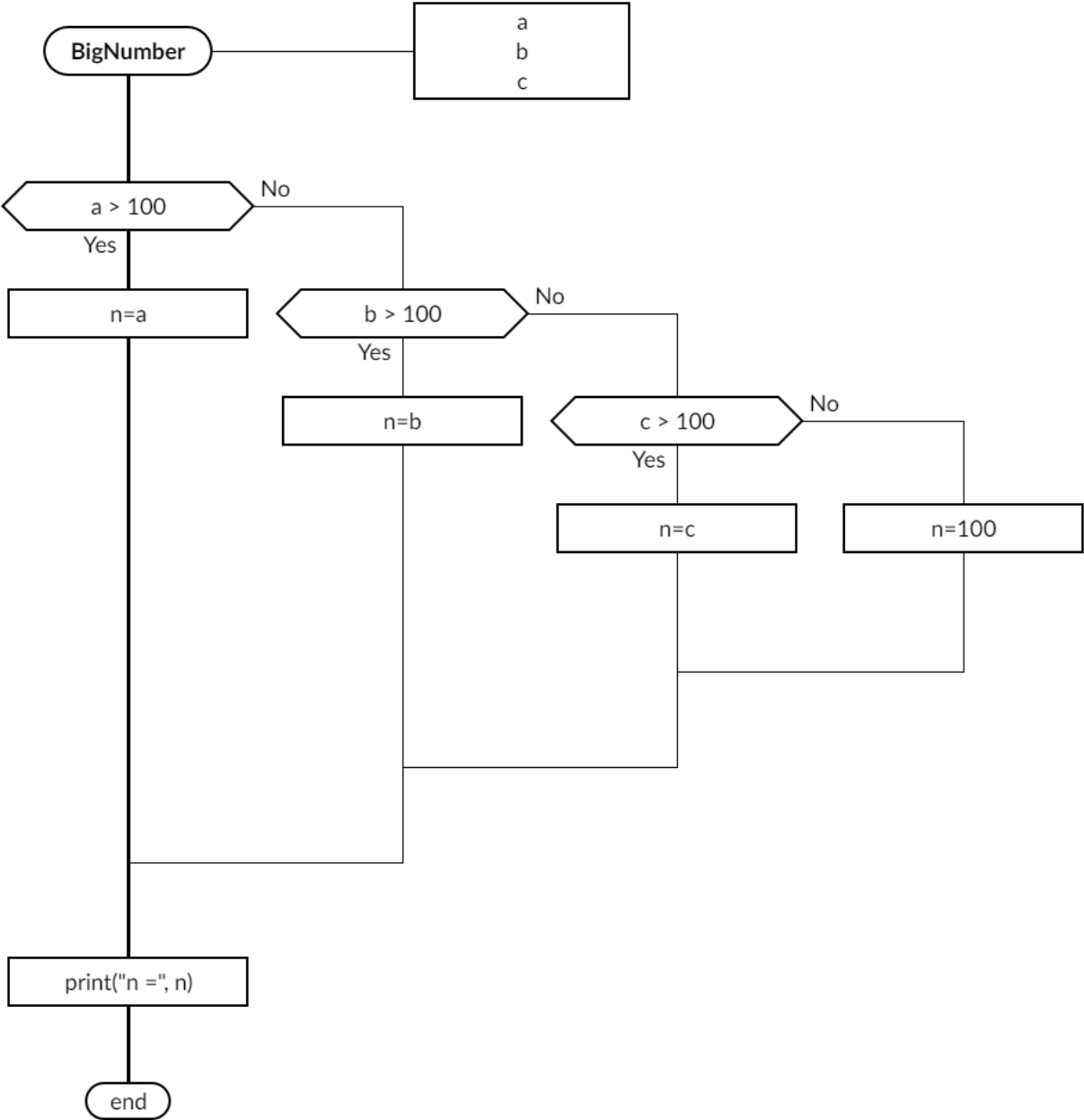
```
def cycie cond():  
    for i in range(1, 10):  
        if i%2==0:  
            print("even:")  
        else:  
            print("not even:")  
    print(i)
```

Original code:

```
def cycle_cond():  
    for i in range(1, 10):  
        if i%2==0:  
            print("even:")  
        else:  
            print("not even:")  
    print(i)
```

Metric: 0.9931428571428572

Файл 20221228203303.png:



Результат работы программы:

Run test 20221228203303.png:

Original code:

Generated code:

```
def BigNumber(a, b, Cc):
```

```
    if a>100:
```

```
        n=a
```

```
    else:
```

```
        if b> 100:
```

```
            n=b
```

```
        else:
```

```
            if c>100:
```

```
                n=c
```

```
            else:
```

```
                n=100
```

```
    print("n =", n)
```

```
def BigNumber(a, b, c):
```

```
    if a>100:
```

```
        n=a
```

```
    else:
```

```
        if b>100:
```

```
            n=b
```

```
        else:
```

```
            if c>100:
```

```
                n=c
```

```
            else:
```

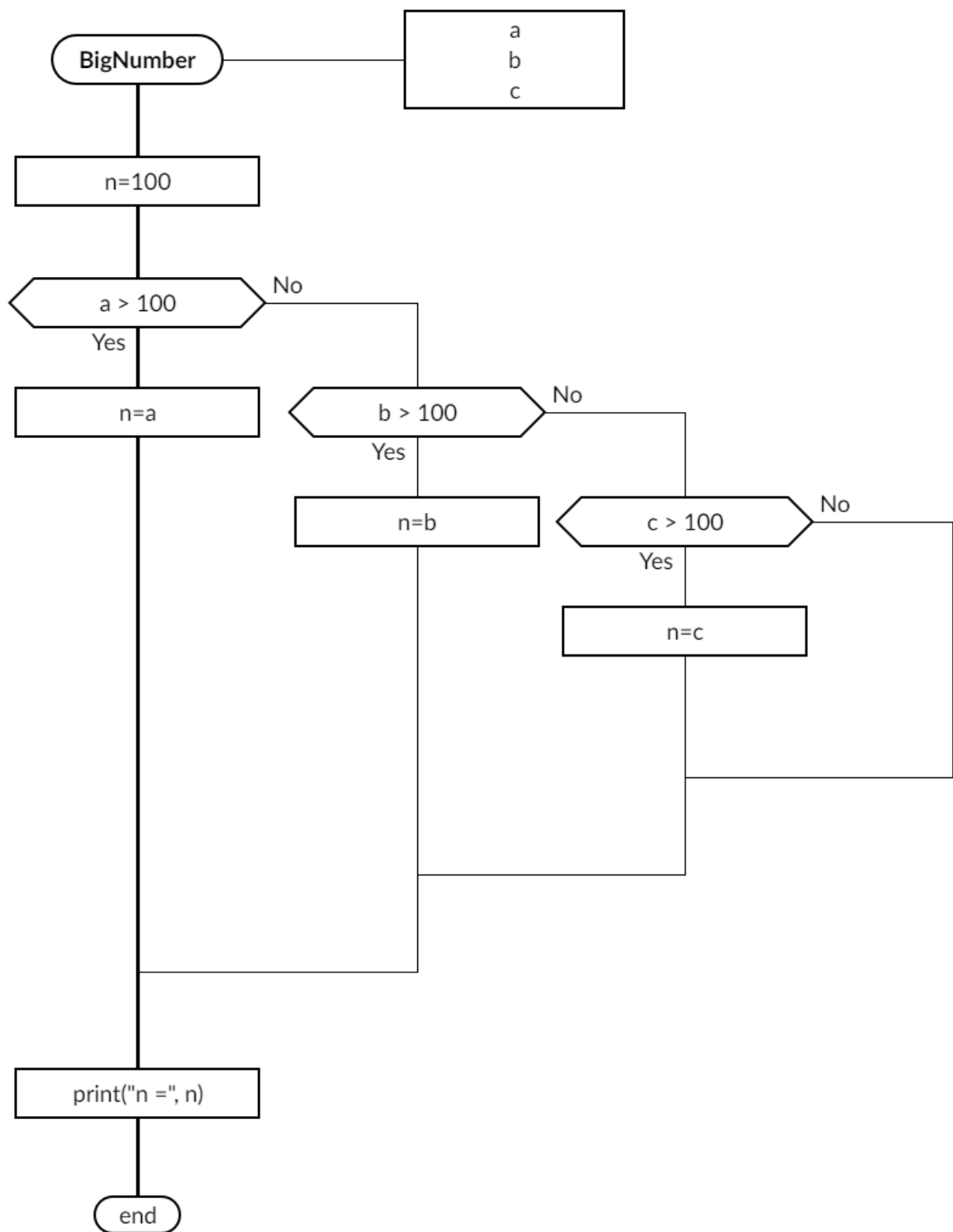
```
                n=100
```

```
    print("n=", n)
```

Metric: 0.997000000000000001

Text detected

Файл 20221228213026.png:



Результат работы программы:

Run test 20221228213026.png:

Generated code:

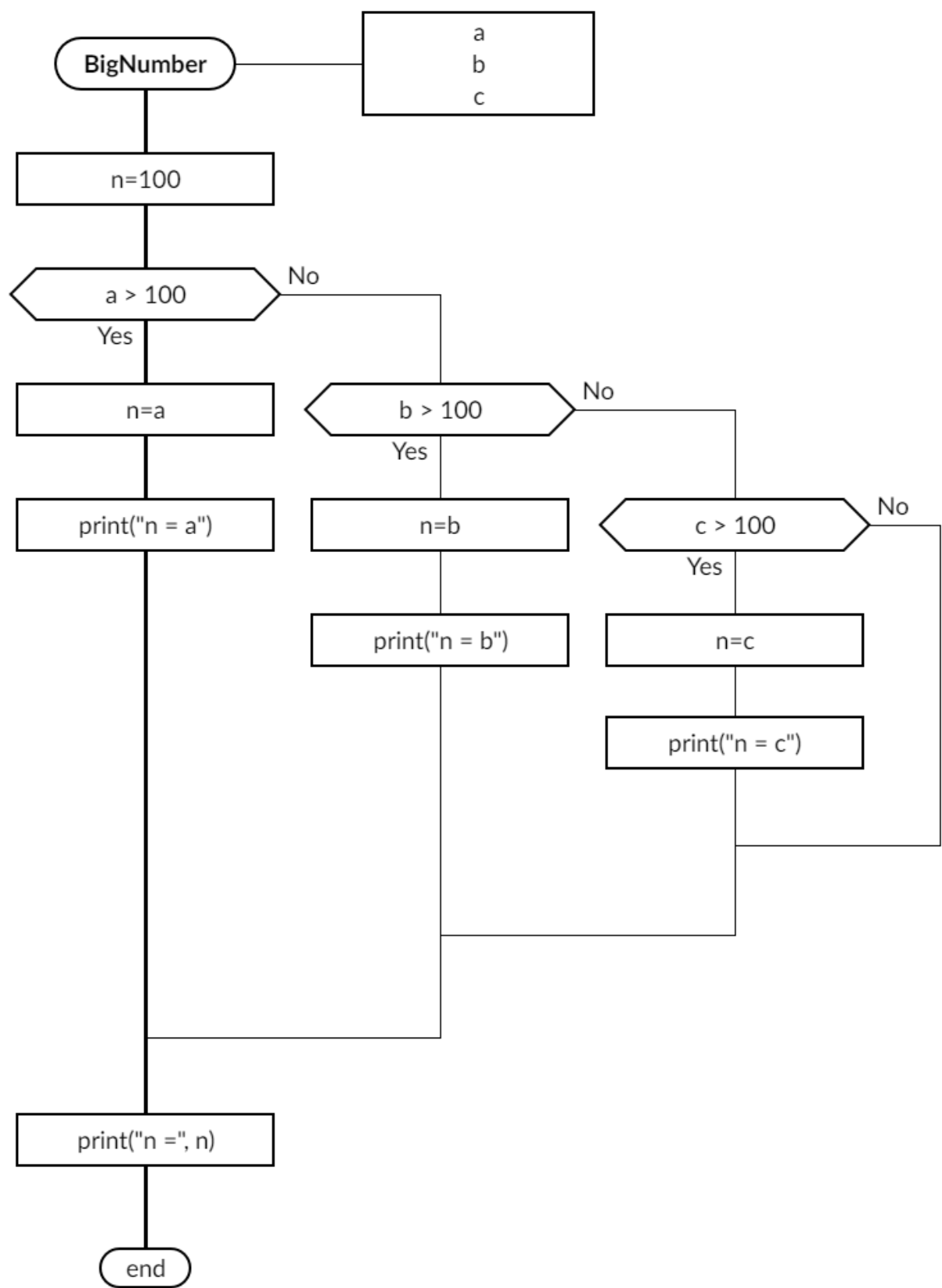
```
def BigNumber(a, b, Cc):  
    n=100  
    if a> 100:  
        n=a  
    else:  
        if b> 100:  
            n=b  
        else:  
            if c> 100:  
                n=c  
    print("n =", n)
```

Original code:

```
def BigNumber(a, b, c):  
    n=100  
    if a>100:  
        n=a  
    else:  
        if b>100:  
            n=b  
        else:  
            if c>100:  
                n=c  
    print("n=", n)
```

Metric: 0.9934545454545456

Файл 20221228223920.png:



Результат работы программы:

Run test 20221228223920.png:

Original code:

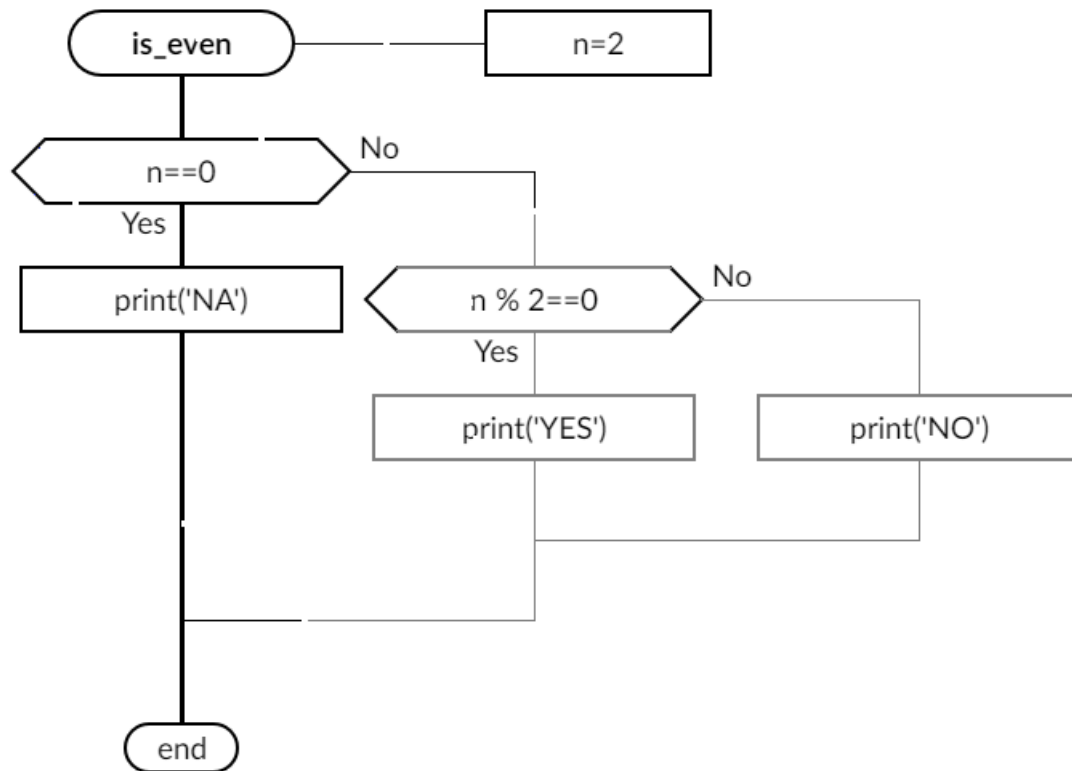
Generated code:

```
def BigNumber(a, b, Cc):
    n=100
    if a>100:
        n=a
        print("n = a")
    else:
        if b> 100:
            n=b
            print("n = b")
        else:
            if c> 100:
                n=c
                print("n = c")
    print("n =", n)
```

```
def BigNumber(a, b, c):
    n=100
    if a>100:
        n=a
        print("n=a")
    else:
        if b>100:
            n=b
            print("n=b")
        else:
            if c>100:
                n=c
                print("n=c")
    print("n=", n)
```

Metric: 0.990857142857143

Файл 20221229140344.png:



Результат работы программы:

Run test 20221229140344.png:

Generated code:

```
def is even(n=2):  
    if n==0:  
        print('NA')  
    else:  
        if n%2==0:  
            print('YES')  
        else:  
            print('NO')
```

Original code:

```
def is_even(n=2):  
    if n==0:  
        print('NA')  
    else:  
        if n%2==0:  
            print('YES')  
        else:  
            print('NO')
```

Metric: 0.9870000000000001

Выводы

В данной работе приведены самые показательные примеры конфигураций, которые может распознавать наша программа.

То есть, это несколько уровней вложенности цикла, несколько уровней вложенности для условий, а также их комбинация.

На довольно простых примерах метрика показала качество 1.0, что говорит о том, что мы достаточно хорошо их распознаем при условии отсутствия видимых дефектов текста и фигур.

Однако, в последнем примере приведено изображение с дефектами, которые можно устранить с помощью морфологического замыкания.

Метрика в данном случае все еще достаточно высока.

То есть, мы не опускались в метрика ниже 0.95, что говорит о том, что все модули программы работают хорошо, кроме может быть распознавания текста, которое можно улучшить.