

WSTĘP DO PROGRAMOWANIA

Laboratorium 3

Pętle

Zadania przygotowały: dr Anna Nenca, Laura Grzonka

Zdania logiczne

Zad. 1. Zgadnij, jakie będą wyniki poniższych linijek kodu. Następnie uruchom program i sprawdź swoją intuicję.

```
print(True)
print(Frue)
print(0 == 1)
print(3.14 > 2.42)
print(-1 <= 3)
print(3 != 3)
print(5 != 3)
print(True or False)
print(True or True)
print(True and False)
print(False and False)
print(False or False)
print(1 > 2 or 5 >= 1)
print(1 > 2 and 3 > 1)
print(1 == 1 and 0 == 0)
print(0 == False)
print(0.0 == False)
print(0.1 == False)
```

Zad. 2. Zgadnij, jakie będą wyniki poniższych linijek kodu. Następnie uruchom program i sprawdź swoją intuicję.

```
a = 9
b = 2

print(b ** 2 == a)
print(a + 1 % b == 1)
print((a + 1) % b == 4)
print(a * 2 + 2 == 10 * b)
print(a + b > 10 and a - b < 5)
```

W jaki sposób należy zmienić powyższy program tak, by w każdym wierszu otrzymać przeciwną wartość logiczną? Załóż, że a i b są stałe i nie możesz ich zmienić.

Pętla while

Zad. 1. Sprawdź, co robią następujące programy:

```
a)
i = 1
while i < 6:
    print(i)
```

```

        i += 1
    else:
        print("i is no longer less than 6")

```

b)

```

i = 0
while i < 10:
    i += 1
    if i == 5:
        break
    print("Aktualny numer to", i)
print("Jestem poza pętlą")

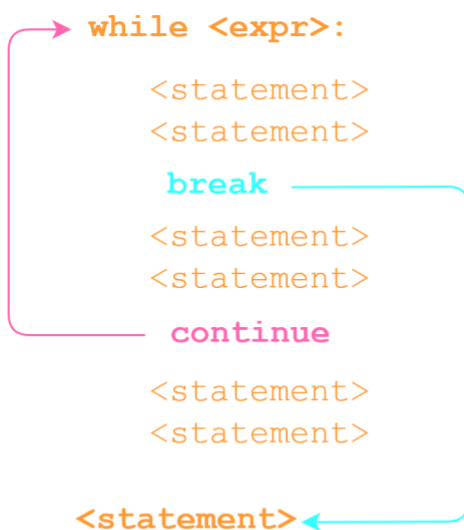
```

c)

```

i = 0
while i < 10:
    i += 1
    if i == 5:
        continue
    print("Aktualny numer to", i)
print("Jestem poza pętlą")

```



Zad. 2. Napisz program, który policzy silnię podanej liczby naturalnej n ($1 * 2 * \dots * n$) przy użyciu pętli while.

Zad. 3. Napisz program, który policzy sumę liczb od 1 do n , gdzie n jest podaną liczbą naturalną przy użyciu pętli while.

Zad. 4. Napisz program, który policzy sumę kwadratów liczb od 1 do n , gdzie n jest podaną liczbą naturalną ($1^2 + 2^2 + \dots + n^2$) (w rozwiązaniu można użyć tylko operatorów porównania, "+", "*" oraz tylko pętli while).

Zad. 5. Napisz program, który policzy sumę sześcianów liczb od 1 do n , gdzie n jest podaną liczbą naturalną ($1^3 + 2^3 + \dots + n^3$) (w rozwiązaniu można użyć tylko operatorów porównania, "+", "*" oraz tylko pętli while).

Zad. 6. Napisz program, który dla zadanej liczby n znajdzie liczbę jej dzielników.

Zad. 7. Napisz program, który dla podanych liczb a i b znajdzie liczbę znaków sumy a+b. (Nie korzystaj z funkcji `len()`)

Zad. 8. Napisz program, który dla zadanej liczby n, następnie wyznaczy największą liczbę m taką, że: $1 + 2 + 3 + \dots + m \leq n$ (przy użyciu pętli while).

Zad. 9. Napisz program, który dla zadanej liczby całkowitej w systemie dwójkowym wyświetli jej odpowiednik w systemie dziesiętnym. Wykonaj to zadanie bez korzystania z wbudowanych funkcji przeliczających na inny system liczbowy.

Zad. 10. Napisz program, który wyświetli wszystkie liczby mniejsze od 100 oraz większe od 1, które przy dzieleniu przez 5 dają resztę 1 oraz przy dzieleniu przez 7 dają resztę 3.

Zad. 11. Napisz program, który będzie sumował liczby parzyste wprowadzone przez użytkownika. Program powinien zatrzymać się, gdy użytkownik wpisze liczbę nieparzystą.

Zad. 12. Moduł random zawiera m.in. funkcje pozwalające „losować” liczby – znajdować liczby pseudolosowe. Korzystając z funkcji `random()` z tego modułu, stwórz grę, w której program losuje liczbę od 1 do 100, a użytkownik ma zgadnąć tę liczbę. Program powinien podpowiadać, czy zgadywana liczba jest za duża, czy za mała, i zakończyć działanie, gdy użytkownik zgadnie poprawnie.

Więcej informacji o tym module:

<http://books.icse.us.edu.pl/runestone/static/thinkcspy/PythonModules/TheRandomModule.html>

Zad. 10. Poproś użytkownika o podanie słowa, a następnie za pomocą pętli while odwróć to słowo (np. „kot” zamieni się na „tok”).

Zad. 11. Napisz program, który wypisze [ciąg Fibonacciego](#), dopóki wartość n-tego elementu nie przekroczy 1000.