**Odpowiedzi – Szkolenie 1–3**

**Odp. 1**

print()

**Odp. 2**

Konkatenacja stringów, czyli inaczej łączenie ciągów tekstowych.

1 + 1 – niemożliwe, wynikiem będzie suma 2;

“str” + 1 – niemożliwe, błąd typu, nie można połączyć typu „str” z „int”;

“str” + “ “ + “str” – możliwe

**Odp. 3**

liczba1 i liczba2 powinny zostać zamienione na typ „int” zaraz po odebraniu wartości od użytkownika. W przypadku zaprezentowanym w zadaniu wynikiem będzie połączenie wartości liczba1 z wartością liczba2, np.

liczba1 = „100”

liczba2 = „200”

Wynikiem sumy jest:100200

**Odp. 4**

nazwa\_psa = “Reksio” - prawidłowa

nazwaKota = “Filemon” - prawidłowa

zmienna = “Devs-Mentoring” – nieprawidłowa, zbyt ogólna i nie wskazująca jaką daną zawiera

płeć = “mężczyzna” – nieprawidłowa, zawiera polskie znaki

WartoscLiczbowa = 5 – nieprawidłowa, nazwa zmiennej zaczyna się z dużej litery a powinna z małej

czyZonaty = False – prawidłowa

czy LubiPsy = False – nieprawidłowa, nazwa zmiennej musi być jednym ciągiem znaków

**Odp. 5**

z = 5, z = -1, z = 0 – zmienne typu integer

z = 0.0, z = -1.123 – zmienne typu float

z = “c”, z = ‘c’, z = “programowanie” – zmienne typu string

z = False, z = True – zmienne typu boolean

**Odp. 6**

a += 5 -–> a = a + 5 – prawidłowy

a -= 5 -–> a = 5 – a – nieprawidłowy

a \*= 5 -–> a = 5 \* a – nieprawidłowy

a\*= 5 -–> a = a \* 5 – prawidłowy

a /= 5 --> a = 5/a – nieprawidłowy

**Odp. 7**

age = 44

height = int(input(‘Podaj swój wzrost: ‘))

weight = int(input(‘Podaj swoją wagę: ‘))

bmi = weight / height\*\*2

print(f‘Twój indeks BMI wynosi: {bmi}’)

**Odp. 8**

5 / 2 = 2.5 – float

1 + 1 = 2 – integer

1.0 + 1 = 2.0 – float

2.0 + 2.0 = 4.0 – float

“napis” + 1 – błąd typu, próba połączenia typu ‘str’ z ‘int’

“napis” + “napis” = ‘napisnapis’ – string

True + True = 2 – integer

5 % 2 = 1 – integer

10 % 2 = 0 – integer

20 % 30 = 20 – integer

100 % 3 = 1 – integer

100 // 3 = 33 – integer

5 // 3 = 1 – integer

1 // 3 = 0 – integer

3\*\*3 = 27 – integer

2\*\*2 = 4 – integer

2\*\*0 = 1 – integer

**Odp. 9**

text = input(‘Podaj dowolny tekst: ‘)

modified\_text = text[-1] + text[1:-1] + text[0]

print(f‘Zmodyfikowany tekst: {modified\_text}’)

**Odp. 10**

a = 5

b = 6

str(a) + str(b)

**Odp. 11**

person1 = “Teacher”

person2 = “Soldier”

person\_concat = person1[:(len(person1) / 2)] + person2[(len(person2) / 2):-1]

**Odp. 12**

tekst1 = tekst2[:] – przypisanie w całości wartości zmiennej tekst2 do zmiennej tekst1

**Odp. 13**

Ponieważ kod maszynowy zrozumiały dla komputera to ciąg 0 i 1, więc każdy znak naszego kodu źródłowego, w procesie interpretacji, jest zamieniany na wartość binarną zrozumiałą dla procesora

**Odp. 14**

Szyfrowanie polega na wykonywaniu szeregu operacji arytmetycznych np. dodawanie, dzielenie, mnożenie, itp.

Szyfr Cezara – liczba kluczy 25, w celu zaszyfrowania wiadomości poszczególne litery wiadomości są przesuwane o wartość klucza i zastępowane literą wynikającą z przesunięcia o wartość klucza, np. litera szyfrowana: A, klucz 5, nowa litera: F.

RSA – asymetryczny algorytm z kluczem publicznym oparty na liczbach pierwszych, polegający na wybraniu losowo dwóch, dość dużych i zbliżonych do siebie pod względem postaci binarnej liczb pierwszych ‘p’ i ‘q’, następnie obliczamy ich iloczyn ‘n’ i obliczamy ‘ʎ’, czyli najmniejszą wspólną wielokrotność NWW liczb ‘p – 1’ i ‘q – 1’. Kolejnym krokiem jest wybranie liczby ‘e’ takiej, która zawiera się pomiędzy ‘1’ a ‘ʎ’ (1 < e < ʎ) i znalezienie liczby ‘d’, która jest odwrotnością modularną liczby ‘e’, czyli (d \* e ≡ 1 (mod ʎ)). Klucz publiczny jest definiowany wówczas jako para liczb (n, e), a klucz prywatny jako para liczb (n, d).

MD5 – algorytm szyfrujący w jedną stronę, tworzący z ciągu znaków o dowolnej długości 128-bitowy skrót. Wiadomości nie da się odszyfrować. Ma zastosowanie np. do szyfrowania haseł użytkowników, gdzie hasło podawane przez użytkownika w procesie np. logowania, jest szyfrowane, a następnie porównywane z hasłem zapisanym w bazie danych.