

Praca domowa po zajęciach nr 2

Termin oddania: przed zajęciami nr 3

* Zadanie 1. Silnia.

Napisz program (skrypt w pliku), który:

- poprosi użytkownika o podanie liczby
- wypisze na ekranie iloczyn, który odpowiada silni z tej liczby
- a następnie wypisze znak równości i wartość tej silni

Przykładowy wydruk programu dla liczby 11:

```
1*2*3*4*5*6*7*8*9*10*11 = 39916800
```

Podpowiedź: należy skorzystać z pętli „while”.

* Zadanie 2. Ping-pong.

Napisz program (skrypt w pliku), który wypisze liczby od 1 do 100, przy czym przy każdej liczbie podzielnej przez 3 napisze słowo: „ping”, przy każdej podzielnej przez 5 napisze „pong”, a przy każdej liczbie podzielnej i przez 3, i przez 5, napisze słowo: „ping-pong” (już bez „ping” i bez „pong”).

Każda liczba ma się znaleźć w osobnej linii. Sformatuj wyświetlane napisy tak, by liczba ze słowem „ping” pojawiała się przy lewej krawędzi ekranu, liczba i „ping-pong” na środku, a liczba i „pong” – wyraźnie bliżej prawej krawędzi. Przejrzyj wyniki działania programu.

Czy widzisz regularność wzoru? Skąd ona się bierze?

Podpowiedź: przyjmijmy, że ekran ma szerokość 72 znaków. Aby wypisać 10 kolejnych znaków spacji (puste miejsce), a następnie dowolny ciąg znaków, należy napisać:

```
print ' '*10 + 'tutaj znaki'
```

Teraz już łatwo można obliczyć ile spacji przed napisem „pong”, a ile przed napisem „ping-pong”, musi znaleźć się na ekranie.

** Zadanie 3. Rozkład na czynniki pierwsze.

Napisz program (skrypt w pliku), który:

- o poprosi użytkownika o podanie liczby,
- wypisze tę liczbę, znak równości, a następnie jej rozkład na czynniki pierwsze

Poprawnym rozwiązaniem dla liczby 120 będzie zatem wydruk:

```
120 = 2 * 2 * 2 * 3 * 5
```

Załącz, że użytkownik będzie wpisywał tylko liczby, ale nie zakładaj, że będą to zawsze liczby poprawne. W przypadku wpisania liczby, której nie da się rozłożyć na czynniki pierwsze niech Twój program wyświetli odpowiedni komunikat o błędzie.

Podpowiedź 1: rozwiązanie tego zadania wymaga użycia listy dzielników:

```
for dzielnik in [2,3,5,7,11,13,.....]:  
    # tutaj algorytm sprawdzania czy dana liczba jest rzeczywiście dzielnikiem
```

Dopuszczalne są również rozwiązania (choć co najwyżej za dwa punkty), w których obsługiwanych będzie co najmniej dziesięć najmniejszych możliwych czynników, a w niektórych przypadkach program na ostatniej pozycji wyświetli niepodzielony iloczyn pewnej liczby nieobsługiwanych czynników wraz z komentarzem:

74730 = 2*3*5*2491 (ostatnia liczba być może wymaga dodatkowego dzielenia)

podczas gdy pełen rozkład na czynniki ma postać:

74730 = 2*3*5*47*53

Podpowiedź 2: pętla może mieć również taką postać:

```
while True:  
    # tutaj kod algorytmu  
    if warunek == True:  
        break
```