

ZAPISYWANIE ROZWIĄZANIA PROBLEMU ITERACYJNEGO.

Przykład 1.

Algorytm sumowania n liczb całkowitych wprowadzanych z klawiatury na podstawie specyfikacji i listy kroków poniżej.

Zadanie: Oblicz sumę n liczb całkowitych.

Dane: liczba naturalna n większa od zera, oznaczająca, ile liczb ma być sumowanych, n dowolnych liczb całkowitych.

Wynik: wartość sumy: **suma**.

Lista kroków:

1. Zaczynaj algorytm.
2. Zmiennej **suma** przypisz wartość 0.
3. Zmiennej **i** przypisz wartość 0.
4. Podaj, ile liczb ma być zsumowanych. Zapamiętaj wprowadzoną liczbę w zmiennej **n**.
5. Wprowadź liczbę do sumowania – zapamiętaj ją w zmiennej **a**.
6. Zmiennej **suma** przypisz wartość zmiennej **suma** powiększoną o wartość zmiennej **a**: **suma = suma + a**.
7. Zmiennej **i** przypisz wartość zmiennej **i** (kolejna wprowadzona liczba) powiększoną o 1: **i = i + 1**.
8. Jeśli **i < n**, wróć do kroku 5.
9. Wyprowadź wynik: **suma**.
10. Zakończ algorytm.

Przypisanie **suma = suma + a** oznacza „pod zmienną **suma** podstaw poprzednią wartość **suma** zwiększoną o wartość kolejnej liczby **a**”.

suma = 4

suma = suma + 40

Wynikiem będzie 44

```
n = int(input("Ile liczb: "))

suma = 0
for i in range(n):
    a = int(input("Wprowadź liczbę: "))
    suma += a #to samo co zapis: suma = suma + a
print("Suma wynosi:", suma)

input("\n\nAby zakończyć, naciśnij Enter")
```

Operator inkrementacji i dekrementacji

Rodzaj operatora	Przykład zastosowania	Odpowiednik	Znaczenie
Operator inkrementacji	$i += 1$	$i = i + 1$	Przypisanie wartości zmiennej i poprzedniej wartości tej zmiennej powiększonej o 1
Operator dekrementacji	$i -= 1$	$i = i - 1$	Przypisanie wartości zmiennej i poprzedniej wartości tej zmiennej pomniejszonej o 1

Ćwiczenie 1.

Zapisujemy algorytm sumowania n liczb w języku programowania.

1. Przepisz program z **Przykładu 1**.
2. Zapisz program w pliku pod nazwą **suma_n_a**.
3. Uruchom i przetestuj program dla kilku różnych wartości zmiennych.

Ćwiczenie 2.

1. Do programu **suma_n_a** dodaj możliwość sprawdzania poprawności wprowadzanej wartości zmiennej n , zgodnie ze specyfikacją zadania w **Przykładzie 1** (liczba naturalna n większa od zera). Sumowanie wprowadzanych liczb powinno być wykonywane dla poprawnej wartości n . Jeśli użytkownik wprowadzi błędne dane, powinien wyświetlić się komunikat „Niepoprawne dane. Liczba powinna być większa od zera”.
2. Zapisz plik pod nazwą **suma_n_b**
3. Uruchom program i przetestuj dla różnych wartości zmiennej n .

Ćwiczenie 3.

1. Zmodyfikuj program **suma_n_b** tak aby sumował tylko wprowadzone przez użytkownika liczby parzyste
2. Zapisz plik pod nazwą **suma_n_b_parz**

Ćwiczenie 4.

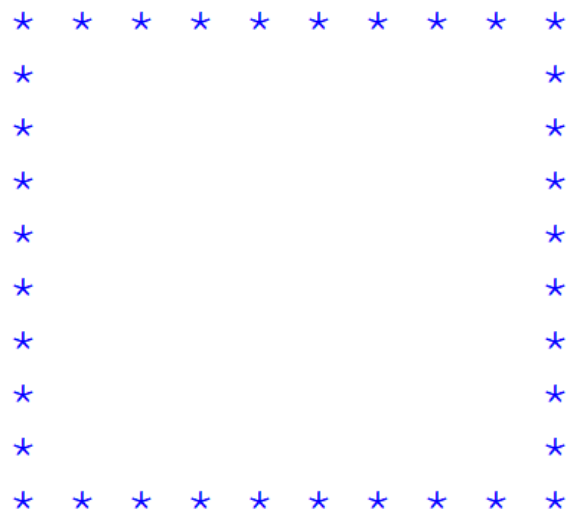
1. Napisz specyfikację zadania i program obliczający sumę n kolejnych liczb naturalnych począwszy od cyfry 1. Program powinien sprawdzać poprawność wprowadzania wartości zmiennej n informując o ew. błędnie wpisanej ilości liczb.
2. Zapisz program w pliku pod nazwą **kolejna_suma_n**.
3. Uruchom i sprawdź jego działanie dla różnych wartości zmiennej n
Podpowiedź: użyj do wykonania zadania funkcję `range(początek, koniec)`

Ćwiczenie 5.

1. Napisz program realizujący algorytm obliczania iloczynu n dowolnych liczb całkowitych. Wynik iloczynu zapamiętuj w zmiennej iloczyn i wyprowadź na ekran.
2. Zapisz program w pliku pod nazwą **iloczyn_n**.
Podpowiedź: Zmiennej iloczyn na początku programu przypisz wartość 1.

Ćwiczenie 6.

1. Napisz program wykonujący poniższą grafikę:



2. Zapisz program w pliku pod nazwą **gwiazdki**