Warunki jakie powinno spełniać rozwiązanie

1. W implementacji można korzystać tylko z elementarnych konstrukcji Python'a, takich jak: funkcje, rekurencja instrukcje warunkowe, pętle.

2. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad5.py`. Krótki opis rozwiązania powinien być umieszczony na początku pliku w formie komentarza (w pierwszej linii w komentarzu powinno być imię i nazwisko studenta). Niedopuszczalne jest w szczególności:

1. zmienianie nazwy funkcji implementującej algorytm, listy jej argumentów, lub nazwy pliku z rozwiązaniem,

2. wypisywanie na ekranie jakichkolwiek napisów, ew. napisy dodane na potrzeby diagnozowania błędów należy usunąć przed wysłaniem zadania.

Zadania niezgodne z powyższymi ograniczeniami otrzymają ocenę 0 pkt. Rozwiązania w innych formatach (np. `.PDF`, `.DOC`, `.PNG`, `.JPG`) z definicji nie będą sprawdzane i otrzymają ocenę 0 punktów, nawet jeśli będą poprawne.

Czas na rozwiązanie zadania 25 minut.
Oceniane będą: czytelność, poprawność i efektywność rozwiązań.

Zadanie 5

Dana jest tablica $T[N]$ wypełniona niepowtarzającymi się liczbami naturalnymi. Proszę zaimplementować funkcję `trojki(T)` która zlicza wszystkie trójki liczb, które spełniają następujące warunki:
(1) największym wspólnym dzielnikiem trzech liczb jest liczba 1,
(2) pomiędzy dwoma kolejnymi elementami trójki może być co najwyżej jedna przerwa.
Funkcja powinna zwrócić liczbę znalezionych trójek.

Przykładowe wywołania funkcji:

```
print(trojki([2,4,6,7,8,10,12])) # 0 trójek
print(trojki([2,3,4,6,7,8,10])) # 1 trójka (3,4,7)
print(trojki([2,4,3,6,5])) # 2 trójki (2,3,5),(4,3,5)
print(trojki([2,3,4,5,6,8,7])) # 5 trójek (2,3,5),(3,4,5),(3,5,8),(5,6,7),(5,8,7)
```

Warunki jakie powinno spełniać rozwiązanie

1. W implementacji można korzystać tylko z elementarnych konstrukcji Python'a, takich jak: funkcje, rekurencja instrukcje warunkowe, pętle.

2. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad6.py`. Krótki opis rozwiązania powinien być umieszczony na początku pliku w formie komentarza (w pierwszej linii w komentarzu powinno być imię i nazwisko studenta). Niedopuszczalne jest w szczególności:

1. zmienianie nazwy funkcji implementującej algorytm, listy jej argumentów, lub nazwy pliku z rozwiązaniem,

2. wypisywanie na ekranie jakichkolwiek napisów, ew. napisy dodane na potrzeby diagnozowania błędów należy usunąć przed wysłaniem zadania.

Zadania niezgodne z powyższymi ograniczeniami otrzymają ocenę 0 pkt. Rozwiązania w innych formatach (np. `.PDF`, `.DOC`, `.PNG`, `.JPG`) z definicji nie będą sprawdzane i otrzymają ocenę 0 punktów, nawet jeśli będą poprawne.

Czas na rozwiązanie zadania 25 minut.
Oceniane będą: czytelność, poprawność i efektywność rozwiązań.

Zadanie 6

Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node
..def init(self,val):
....self.val = val
....self.next = None
```

Zbiór mnogościowy liczb naturalnych reprezentowany jest przez listę o uporządkowanych rosnąco elementach. Proszę napisać funkcję `iloczyn(z1,z2,z3)`, która przekształca 3 listy (zbiory) z1,z2,z3 w jedną listę (zbiór) zawierającą elementy będące częścią wspólna zbiorów z1,z2,z3. Funkcja powinna zwrócić wskazanie do listy wynikowej.

Komentarz: Zadanie jest tak proste, że nie wymaga przykładu ani danych testowych.