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.
3. Niedopuszczalne jest używanie typu str.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad4.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 3

Dana jest tablica $T[N][N]$ (reprezentująca szachownicę) wypełniona liczbami całkowitymi. Proszę zaimplementować funkcję `chess(T)` która ustawia na szachownicy dwie wieże, tak aby suma liczb na „szachowanych" przez wieże polach była największa. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić położenie wież w postaci krotki $(row_1, col_1, row_2, col_2)$.
Uwaga: zakładamy, że pola na których znajdują się wieże nie są szachowane.

Przykładowe wywołania funkcji:

```
print(chess([[4,0,2],[3,0,0],[6,5,3]])) # (0,1,1,0) suma=17
print(chess([[1,1,2,3],[-1,3,-1,4], [4,1,5,4], [5,0,3,6]] # (2,3,3,1) suma=35
```

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.
3. Niedopuszczalne jest używanie typu str.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad4.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 4

Dana jest liczba naturalna N. Proszę zaimplementować funkcję `divide(N)`, która sprawdza czy jest możliwe pocięcie liczby N na kawałki, tak aby każdy z kawałków był liczba pierwszą oraz liczba kawałków też była liczbą pierwszą. Funkcja powinna zwracać wartość logiczną. Na przykład: `divide(2347)=True`, podział na 23 i 47, natomiast `divide(2255)=False`.

Przykładowe wywołania funkcji:

```
print(divide(273)) # True, podział 2|7|3
print(divide(22222)) # True, podział 2|2|2|2|2
print(divide(23672)) # True, podział 23|67|2
print(divide(2222)) # False
print(divide(21722)) # False
```