Warunki jakie powinno spełniać rozwiązanie

1. W implementacji można korzystać tylko z elementarnych konstrukcji Python'a, takich jak: funkcje, instrukcje warunkowe, pętle.
2. Nie wolno korzystać z wbudowanych algorytmów sortowania.
3. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad1.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 rozwiazań.

Zadanie 1

Napis nazywamy wielokrotnym, jeżeli powstał przez n-krotne ($n > 1$) powtórzenie innego napisu o długości co najmniej 1. Przykłady napisów wielokrotnych: $ABCABCABC$, $AAAA$, $ABAABA$. Dana jest tablica $T[N]$ zawierająca napisy. Proszę napisać funkcję `multi(T)`, która zwraca długość najdłuższego napisu wielokrotnego występującego w tablicy T lub wartość 0, jeżeli takiego napisu nie ma w tablicy.

Warunki jakie powinno spełniać rozwiązanie

1. W implementacji można korzystać tylko z elementarnych konstrukcji Python'a, takich jak: funkcje, instrukcje warunkowe, pętle.
2. Należy założyć, że typ int jest ograniczony do liczb 64 bitowych
3. Nie wolno korzystać z typu str.
4. Nie wolno korzystać z wbudowanych algorytmów sortowania.
5. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych.

Format rozwiązań

Implementacja funkcji powinna się znajdować w pliku o nazwie `zad2.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 rozwiazań.

Zadanie 2

Dana jest tablica $T[N][N]$ wypełniona wartościami $0, 1$. Każdy wiersz tablicy traktujemy jako liczbę zapisaną w systemie dwójkowym o długości N bitów. Stała N jest rzędu 1000. Proszę zaimplementować funkcję `distance(T)`, która dla takiej tablicy wyznaczy dwa wiersze, dla których różnica zawartych w wierszach liczb jest największa. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić odległość pomiędzy znalezionymi wierszami. Można założyć, że żadne dwa wiersze nie zawierają identycznego ciągu cyfr.