

```

Zdefiniuj bombel(warta):
    _ n = ile elementow na warta
    _ dla i jest w zakresie do n:
        _ dla j jest w zakresie od 0 do n-i-1:
            _ jezeli warta od i jest wieksza od wart y od j+1 to:
                _ warta od j, warta od j+1 = warta od j+1, warta od j
    _ Zwróc warta
    _ Warta = tablica
    Drukuj bombel od wart y

Zdefiniuj scalenko od wart y
    _ jezeli ilosc elementow w wartce jest <=1:
        _ zwróc wart e
    _ srodek = ilosc el w wartce // 2
    _ Lewa polowa = scalenko od wart y[:srodek]
    _ Prawa polowa = scalenko od wart y[srodek:]
    _ Zwróc sortki od lewej polowy, prawej polowy
    Zdefiniuj sortki od lewej, prawej:
        _ Wynik = []
        _ i = 0
        _ dopoki i < liczby elementow lewego oraz j < liczby elementow prawego
            _ jezeli lewy od i < prawy od j:
                _ wynik.append(lewy od i)
                _ i += 1
            _ w przeciwnym wypadku:
                _ wynik.append(prawy od j)
                _ j += 1
        _ wynik.extend(lewy od i[:])
        _ wynik.extend(prawy od j[:])
        _ zwróc wynik
    Warta = tablica
    Drukuj scalenko od wart y

Zdefiniuj frankenstein od n
    _ n = calkowite (n)
    _ jezeli n <= 0
        _ zwróc 0
    _ jezeli nie to czy n == 1:
        _ zwróc 1
    _ W przeciwnym wypadku :
        _ zwróc frankenstein(n-1) + frankenstein(n-2)
    Ktory = calkowita
    Drukuj frankenstein od ktory

Zdefiniuj sortowy bieranie od lista
    _ n = ilosc elementow od lista
    _ dla i w zakresie n:
        _ minindex = i
        _ dla j w zakresie i+1, n:
            _ jezeli lista od j < lista[minindex]:
                _ minindex = j
    _ lista od i, lista od minindex = lista od minindex, lista od i
    _ Zwróc lista

Cezar
    Pobierz tekst.
    Pobierz przesuniecie i wprowadz.

```

```

Przesuniecie zamien na liczbe calkowita za pomoca funkcji i to zapisz
wartosc w zmiennej.
Utworz pusty ciaz znakow bedacych wynikiem.
Dla znak w teksc:
    _ Sprawdzamy jesli znak jest litero.
    _ Wtedy kod = ord(znak).
    _ przesuniecie bazowe = 65 jesli duza, w przeciwnym wypadku 97.
    _ szyfrowanie = (kod - przesuniecie bazowe + przesuniecie) reszta z
    _ dzielenia przez 26 i dodajemy przesuniecie bazowe
    _ Wynik rowny dodać chr(szyfrowanie)
    w przeciwnym wypadku:
        _ wynik plus rowna sie znak
        _ Wyswietl komunikat i wynik.

Dziesietna
    definiujemy funkcje dziesietna na binarna (liczba)
    _ jesli liczba rowna rowna zero
        _ zwracamy "zero"
    _ tworzymy pusty ciaz znakow binarna
    _ dopoki liczba wieksza od zero
        _ binarna rowna stro (reszta z dzielenia liczby przez 2) plus binarna
        _ liczba dzielmy dzielmy rowne 2
    _ zwracamy binarna
    definiujemy dziesietna na szesnastkowa (liczba)
    _ jesli liczba rowna rowna zero
        _ zwróc "zero"
    _ pusty ciaz szesnastkowa
    _ znaki "0123456789ABCDEF"
    _ dopoki liczba > 0:
        _ szesnastkowa rowna będzę znaki [liczba reszta z dzielenia przez 16] +
        _ szesnastkowa
    _ liczba dzielmy dzielmy rowna 16
    _ zwróc szesnastkowa
    definiuj funkcje maino
    _ liczba i to tekst podaj liczbe
    _ binarna = dziesietna_na_binarna(liczba)
    _ szesnastkowa = dziesietna_na_szesnastkowa(liczba)
    wypisz
    funkcja glowna wywołajo

pierwsza
    definiujemy pierwsza liczba
    _ jesli liczba mniejsza rowna 1:
        _ zwracamy falsz
    _ dla i w zakresie od (2 do pierwsza (i to liczba potega 0.5) plus jeden):
        _ jesli liczba dzieli sie przez i rowne rowne zero
            _ zwraca falsz
        _ zwraca prawda
    _ zwróc prawda
    zdefiniuj parzysta liczba:
        _ zwróc reszta z dzielenia liczba przez 2 rowna rowna zero
    _ liczba i to wpiszy pytanie o liczbe
    _ jesli parzysta i pierwsza:
        _ wypisz
    _ jezeli nie to czy * 2
    _ inaczej nie jest parzysta ani pierwsza
    Franek

```

```

Zdefiniuj frankenstein od n
    _ jezeli n == 0
        _ zwróc zero
    _ jezeli nie to czy n == 1
        _ zwróc 1
    _ W przeciwnym razie
        _ zwróc frankenstein(n-1) + franek(n-2)
    N i to wpiszy pytanie o to ktory wyraz ciazu
    _ jezeli n < 0 wpiszy ze niezdefiniowany dla ujemnych
    W przeciwnym wypadku
        _ Wynik = franek
        _ Podaj odpowiedz

Zdefiniuj konwerter od liczba:
    _ jezeli liczba < 0
        _ zwróc "bladdd"
    _ binarna to bin od (liczba)[2:]
    _ szesnastkowa = hex(liczba)[2:] duza
    _ zwróc binarna, szesnastkowa
    _ wejście to liczba calkowita
    _ drukuj konwertera od wejścia

Jesli chcesz wrócić to
    Liczba dziesietna jest rowna calkowita(binarna, 2) itd.

Zdefiniuj pierwsza (n):
    _ dla i w zakresie (2, n)
        _ jezeli n reszta z dzielenia przez i rowne rowne 0:
            _ zwróc falsz
        _ zwróc prawda

Zdefiniuj listr(napis)
    _ strouit rowny pusty ciaz
    _ dla j w napis:
        _ jesli j w ["a", "b", "r"]:
            _ strouit plus rowny j.upper()
        _ jezeli nie to czy j rowny rowny "q" lub j rowny rowny "Q"
        _ pasuj
    _ w przeciwnym przypadku:
        _ strouit plus rowny j
    _ zwróc strouita
    Lista = ["wpisz kilka slow itd"]
    Lista wyjsciova str = [pusta lista]
    Lista wyjsciova intow = [pusta lista]
    Outputfloat = [pusta lista]

Dla i w lista:
    _ jesli isinstance(i, str):
        _ outputstr.append(listr(i))
    _ jezeli nie to czy isinstance(i, int):
        _ jesli pierwsza(i):
            _ outputfloat.append(i)
        _ jezeli nie to czy isinstance(i, float):
            _ wyprowadz float.append(round(i, 2))
    _ outputstr.append(outputfloat) # Dodaj liste liczb pierwszych do listy napisow
    _ outputstr.append(outputfloat) # Dodaj liste zaokrąglonych liczb do listy
    _ napisow print(outputstr) # Wydrukuj wynik koncowy

```