## Zadanie egzaminacyjne :

## Aplikacja konsolowa

Napisz aplikację konsolową w języku C#, która wykonuje obliczenia potęgi liczby. Program powinien umożliwiać użytkownikowi wprowadzenie podstawy oraz wykładnika, a następnie obliczyć i wyświetlić wynik potęgowania, korzystając z metody szybkiego potęgowania.

**Wymagania:**

1. **Interfejs Użytkownika**:
    - Aplikacja powinna przyjmować od użytkownika dwie wartości:
        - podstawę a (liczba całkowita typu long).
        - wykładnik n (liczba całkowita typu uint).
    - Program powinien wyświetlać wynik w formacie:
        - a do potęgi n wynosi wynik
2. **Algorytm**:
    - Zaimplementuj funkcję rekurencyjną do obliczania potęgi liczby za pomocą szybkiego potęgowania:
        - Jeśli wykładnik n jest równy 0, zwróć 1.
        - Jeśli n jest nieparzyste, zwróć wartość a pomnożoną przez wynik obliczenia potęgi a do n-1.
        - Jeśli n jest parzyste, oblicz wartość w, która jest potęgą a do n/2, a następnie zwróć w * w.
3. **Obsługa Błędów**:
    - Zapewnij, aby użytkownik nie wprowadzał wartości <0 dla wykładnika.
    - Obsłuż sytuacje, w których użytkownik podaje nieprawidłowe dane (np. tekst zamiast liczby).
4. **Testowanie**:
    - Upewnij się, że program działa poprawnie dla różnych przypadków testowych, takich jak:
        - Podstawa 2, wykładnik 10 (2^10 = 1024).
        - Podstawa 2, wykładnik 0 (2^0 = 1).
        - Podstawa 5, wykładnik 1 (5^1 = 5).
        - Podstawa -2, wykładnik 3 (-2^3 = -8)
        - Podstawa 2, wykładnik -1 („wprowadź poprawna wartość)

**Zadanie egzaminacyjne: Obliczanie n-tego wyrazu ciągu rekurencyjnego**

Napisz aplikację konsolową, która będzie obliczać wartość n-tego wyrazu ciągu, zdefiniowanego według następujących reguł:

$$a_n = \begin{cases} 1, & dla\ n = 1 \\ 0,5, & dla\ n = 2 \\ -a_{n-1} \cdot a_{n-2}, & dla\ n > 2 \end{cases}$$

**Wymagania:**

1. **Interfejs Użytkownika**:
   - Program powinien umożliwiać użytkownikowi wprowadzenie liczby całkowitej n, dla której chce obliczyć wyraz ciągu.
   - Program powinien wyświetlić wynik w formacie: n wyraz ciągu ma wartość x
2. **Obsługa Błędów**:
   - Program powinien zapewniać, że użytkownik wprowadza jedynie dodatnie liczby całkowite. W przypadku wprowadzenia nieprawidłowych danych, użytkownik powinien być poproszony o ponowne wprowadzenie wartości.
3. **Testowanie**:
   - Upewnij się, że program poprawnie oblicza wartości dla różnych przypadków testowych, takich jak:
     - n = 1 (wynik: 1)
     - n = 2 (wynik: 0.5)
     - n = 3 (wynik: -0.5)
     - n = 4 (wynik: 0.25)
     - n = 5 (wynik: -0.125)

**Zadanie egzaminacyjne: Obliczanie n-tego wyrazu ciągu rekurencyjnego**

Napisz aplikację konsolową, która będzie obliczać i wypisywać kolejne wyrazy ciągu zdefiniowanego następująco:

$$a_n = \begin{cases} 2, & dla\ n = 1 \\ a_{n-1} * 2, & dla\ n\ nieparzystego \\ a_{n-1} + 2, & dla\ n\ parzystego \end{cases}$$

Dla n= 6 Kolejne wyrazy ciągu to 2, 4, 8, 10, 20, 22

Dla n = 1 K olejne wyrazy ciągu to 2

**Zadanie: Zamiana liczby dziesiętnej na liczbę szesnastkową (metoda rekurencyjna)**

Napisz program w języku C#, który zamienia liczbę zapisaną w systemie dziesiętnym na jej odpowiednik w systemie szesnastkowym. W przeciwieństwie do poprzednich zadań, tym razem użyj metody rekurencyjnej.

Szczegóły

1. Program powinien poprosić użytkownika o wprowadzenie liczby całkowitej w systemie dziesiętnym.
2. Następnie program przekształci wprowadzoną liczbę na format szesnastkowy przy użyciu funkcji rekurencyjnej.
3. Wynik powinien być wyświetlony jako liczba szesnastkowa, w której cyfry większe od 9 zamieniane są na litery od A do F.

Przykład działania programu

Przykład 1:

- **Wejście**: 255
- **Wyjście**: FF

Przykład 2:

- **Wejście**: 1234
- **Wyjście**: 4D2

**Zadanie: Zamiana liczby dziesiętnej na liczbę szesnastkową (bez użycia rekurencji)**

Napisz program w języku C#, który zamienia liczbę zapisaną w systemie dziesiętnym na jej odpowiednik w systemie szesnastkowym. Program powinien działać bez użycia rekurencji.

Szczegóły

1. Program powinien prosić użytkownika o podanie liczby całkowitej w systemie dziesiętnym.
2. Następnie powinien zamienić wprowadzoną liczbę na postać szesnastkową (odpowiednią dla systemu liczbowego o podstawie 16).
3. Program powinien wyświetlić wynik w postaci liczby szesnastkowej, gdzie cyfry większe od 9 zamieniane są na litery od A do F.

## Przykład działania programu

Przykład 1:

- **Wejście**: 255
- **Wyjście**: FF

Przykład 2:

- **Wejście**: 1234
- **Wyjście**: 4D2