Zadanie 1.

Wiązka zadań *Ciągi rekurencyjne*

Dana jest następująca funkcja rekurencyjna:

funkcja *wynik*(*i*)
    jeżeli $i < 3$
        zwróć 1 i **zakończ**;
    w przeciwnym razie
        jeżeli i mod $2 = 0$
            zwróć *wynik*($i - 3$) + *wynik*($i - 1$) + 1
        w przeciwnym razie
            zwróć *wynik*($i - 1$) mod 7

Uwaga: Operator mod oznacza resztę z dzielenia.

1.1.

Uzupełnij poniższą tabelę:

i	**wynik(*i*)**
2	1
3	
4	
5	
6	
7	
8	

1.2.

Wykonaniem elementarnym nazywać będziemy wykonanie *wynik*(0), *wynik*(1) lub *wynik*(2). Natomiast *złożonością elementarną wynik*(i) nazywamy liczbę *wykonań elementarnych* będących efektem uruchomienia *wynik*(i). Złożoność elementarną *wynik*(i) oznaczamy przez $E(i)$.

Na przykład złożoność elementarna *wynik*(4) wynosi $E(4) = 2$, ponieważ wykonując *wynik*(4), wywołamy *wynik*(3) i *wynik*(1) (wykonanie elementarne), a z kolei przy wykonaniu *wynik*(3) wywołamy *wynik*(2) (drugie wykonanie elementarne).

Uzupełnij poniższą tabelę:

i	$E(i)$
0	1
3	1
5	
7	
9	
10	

Okazuje się, że $E(i)$ można opisać rekurencyjnym wyrażeniem, którego niekompletną postać podajemy poniżej. Uzupełnij brakujące miejsca tak, aby $E(i)$ dawało poprawną złożoność elementarną *wynik*(i) dla każdego całkowitego nieujemnego i.

E(0) = E(1) = E(2) = 1
E(i) = E(....................) + E(..............)   dla parzystego $i > 2$
E(i) = E(....................)   dla nieparzystego $i > 2$

1.3.

Naszym celem jest wyznaczenie największej liczby spośród wartości funkcji *wynik*(0), *wynik*(1),…,*wynik*(1000) bez konieczności rekurencyjnego wyznaczania kolejnych wartości. Poniżej prezentujemy niekompletny algorytm realizujący to zadanie.

```
W[0] ← 1
W[1] ← 1
W[2] ← 1
max_wart ← 1
dla i = 3, 4, …, 1 000 wykonuj
        jeżeli i mod 2 = 0
                W[i] ← ..............................................................
        w przeciwnym razie
                W[i] ← ..............................................................
        jeżeli W[i] > max_wart
                ..............................................................
zwróć max_wart
```

Uzupełnij brakujące miejsca w algorytmie tak, aby zwracał on największą liczbę spośród *wynik*(0), *wynik*(1),…,*wynik*(1000).