

PRÓBNY EGZAMIN MATURALNY Z INFORMATYKI

POZIOM ROZSZERZONY ARKUSZ I

STYCZEŃ 2012

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 9 stron (zadania 1 – 3). Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Rozwiązania i odpowiedzi zamieść w miejscu na to przeznaczonym.
3. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
4. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
5. Pamiętaj, że zapisy w brudnopisie nie podlegają ocenie.
6. Wpisz obok zadeklarowane (wybrane) przez Ciebie na egzamin środowisko komputerowe, kompilator języka programowania oraz program użytkowy.
7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w wybranej przez siebie notacji: listy kroków, schematu blokowego lub języka programowania, który wybrałeś/aś na egzamin.
8. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.

WYBRANE:

.....
(środowisko)

.....
(kompilator)

.....
(program użytkowy)

Czas pracy:
90 minut
Liczba punktów
do uzyskania: 20

PESEL

--	--	--	--	--	--	--	--	--	--	--

Zadanie 1. Test (5 pkt)

- a) Różnica $BABA_{16} - ABA_{16}$ równa się:
- ☐ $AC0C_{16}$
 - ☐ 130000_8
 - ☐ B_{16}
 - ☐ 1011000100100010_2
- b) Zasady kompresji danych najlepiej uzasadnia przekształcenie napisu AAABBBCCC do postaci:
- ☐ 3A3B3C
 - ☐ trzy A trzy B trzy C
 - ☐ ABC po trzy sztuki
 - ☐ ABC każdego po trzy
- c) Który matematyk nie kojarzy się z żadnym algorytmem:
- ☐ Euklides
 - ☐ Newton
 - ☐ Horner
 - ☐ Pascal
- d) Który algorytm sortujący (standardowo) nie działa *in situ*, czyli wymaga dodatkowej tablicy w czasie działania:
- ☐ Algorytm bąbelkowy (Bubble Sort)
 - ☐ Sortowanie przez scalanie (Merge Sort)
 - ☐ Sortowanie szybkie (Quick Sort)
 - ☐ Sortowanie przez wstawianie (Insertion Sort)

e) Uporządkuj poniższe złożoności algorytmów w kolejności rosnącej:

1. n – złożoność liniowa
2. n^2 – złożoność kwadratowa
3. $\log(n)$ – złożoność logarytmiczna
4. $n \log(n)$ – złożoność liniowo-logarytmiczna

☐ 1, 2, 3, 4

☐ 3, 1, 4, 2

☐ 4, 3, 2, 1

☐ 3, 4, 1, 2

Punktacja:

Wypełnia egzaminator	Podpunkt:	a)	b)	c)	d)	e)	Razem
	Maksymalna liczba punktów:	1	1	1	1	1	5
	Uzyskana liczba punktów:						

Zadanie 2. Liczby Fibonacciego (6 pkt)

Liczby Fibonacciego są definiowane w następujący sposób:

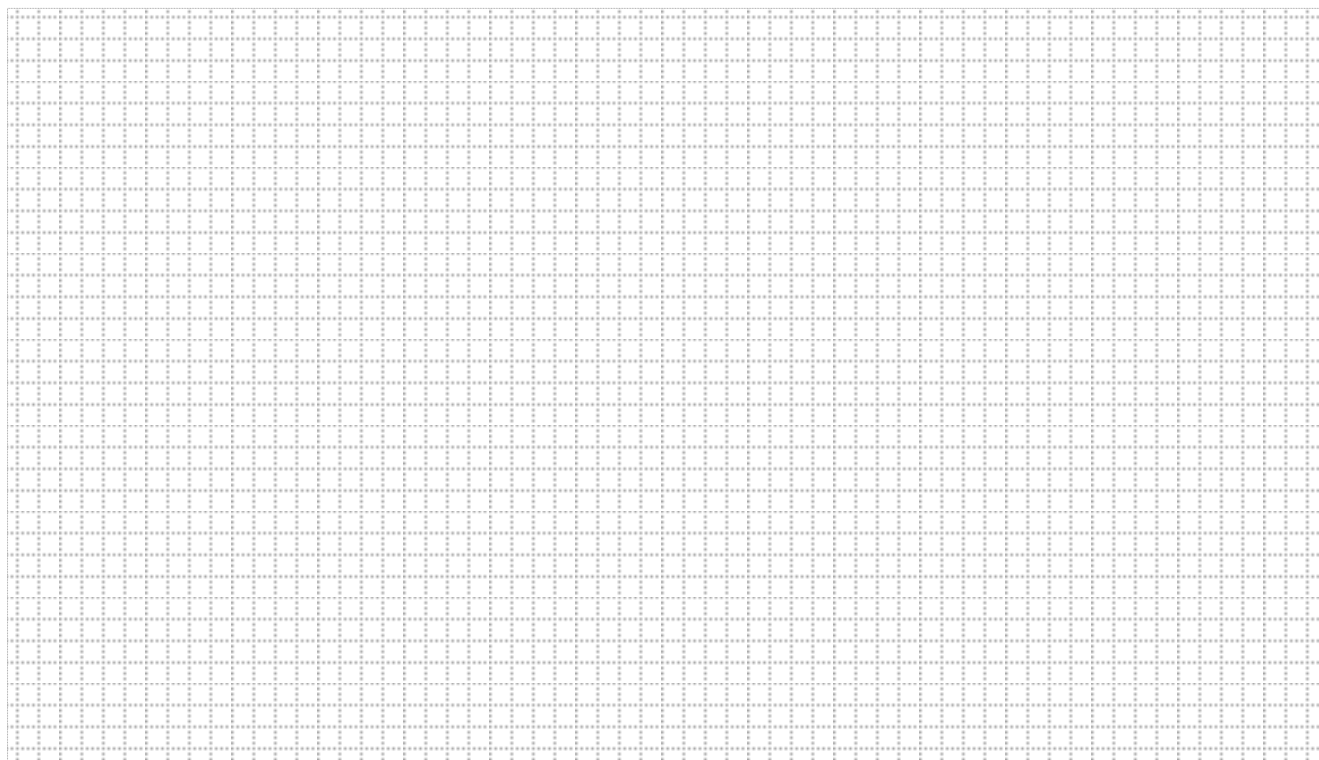
$$F_1 = 1, \quad F_2 = 1,$$

$$F_n = F_{n-1} + F_{n-2} \quad \text{dla } n = 3, 4, \dots$$

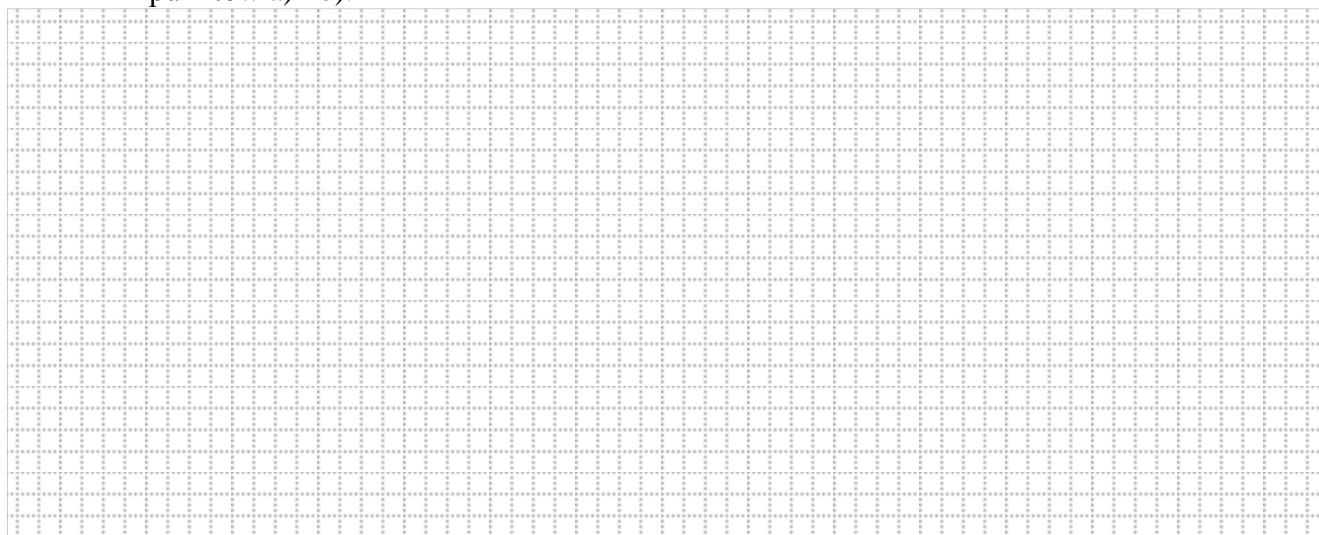
- a) W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) podaj opis rekurencyjnego algorytmu, który służy do obliczania wartości liczby F_n dla dowolnego n .

- b) W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) podaj opis algorytmu, który służy do obliczania wartości liczby F_n dla dowolnego n , ale nie korzysta z rekurencji.

- c) Chcesz obliczyć wartość F_7 . Ile razy podczas obliczania wartości F_7 jest obliczana wartość liczby F_4 , gdy stosujesz algorytm z punktu a), a ile gdy stosujesz algorytm z punktu b)?



- d) Jak zinterpretujesz wyniki otrzymane w punkcie c)? Porównaj działanie algorytmów z punktów a) i b).



Punktacja:

Wypełnia egzaminator	Podpunkt:	a)	b)	c)	d)	Razem
	Maksymalna liczba punktów:	1	2	2	1	6
	Uzyskana liczba punktów:					

Zadanie 3. Progi i schody (9 pkt)

W ciągu liczb naturalnych, parę sąsiednich liczb nazywamy **progiem**, jeśli następna liczba jest mniejsza od poprzedniej.

W ciągu liczb naturalnych, **schody do dołu** tworzy jego podciąg złożony z przynajmniej dwóch liczb, w którym kolejna liczba nie jest większa od poprzedniej i tego ciągu nie można rozszerzyć w jedną albo w drugą stronę do innych schodów do dołu. Liczba elementów w takim podciągu jest **długością schodów**.

Przykład:

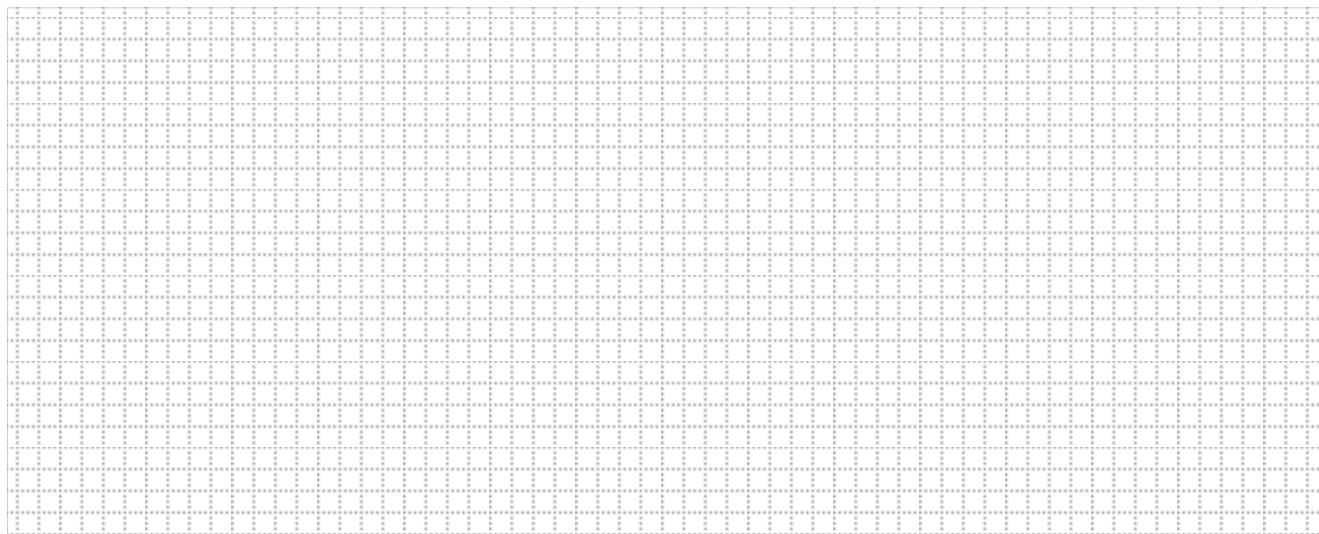
Ciąg: 3, 9, 7, 7, 6, 4, 4, 4, 5 zawiera schody do dołu 9, 7, 7, 6, 4, 4, 4 o długości 7. Te schody do dołu zawierają 3 progi: 9 7, 7 6 i 6 4.

Dane: n i ciąg złożony z n liczb naturalnych

a) Dla następującego ciągu liczb:

2, 2, 2, 3, 1, 1, 3, 3, 1, 10, 11, 7, 7, 6, 7, 7, 8, 9, 9, 7

wypisz kolejno wszystkie występujące w nim schody do dołu i obok każdego schodu podaj ich długości i liczbę progów, jakie zawierają.

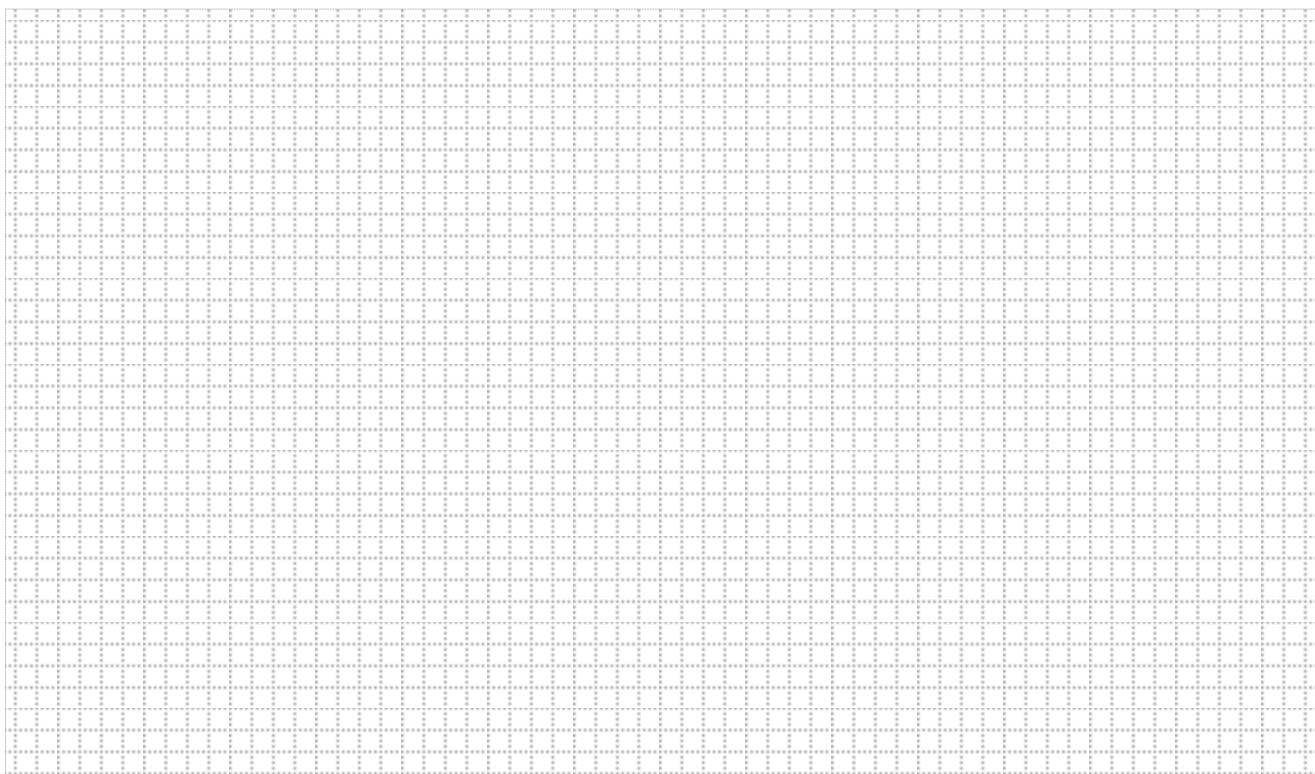


b)

Dane: n – liczba naturalna

ciąg złożony z n liczb naturalnych

W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) podaj opis algorytmu, który oblicza, ile progów znajduje się w danym ciągu.

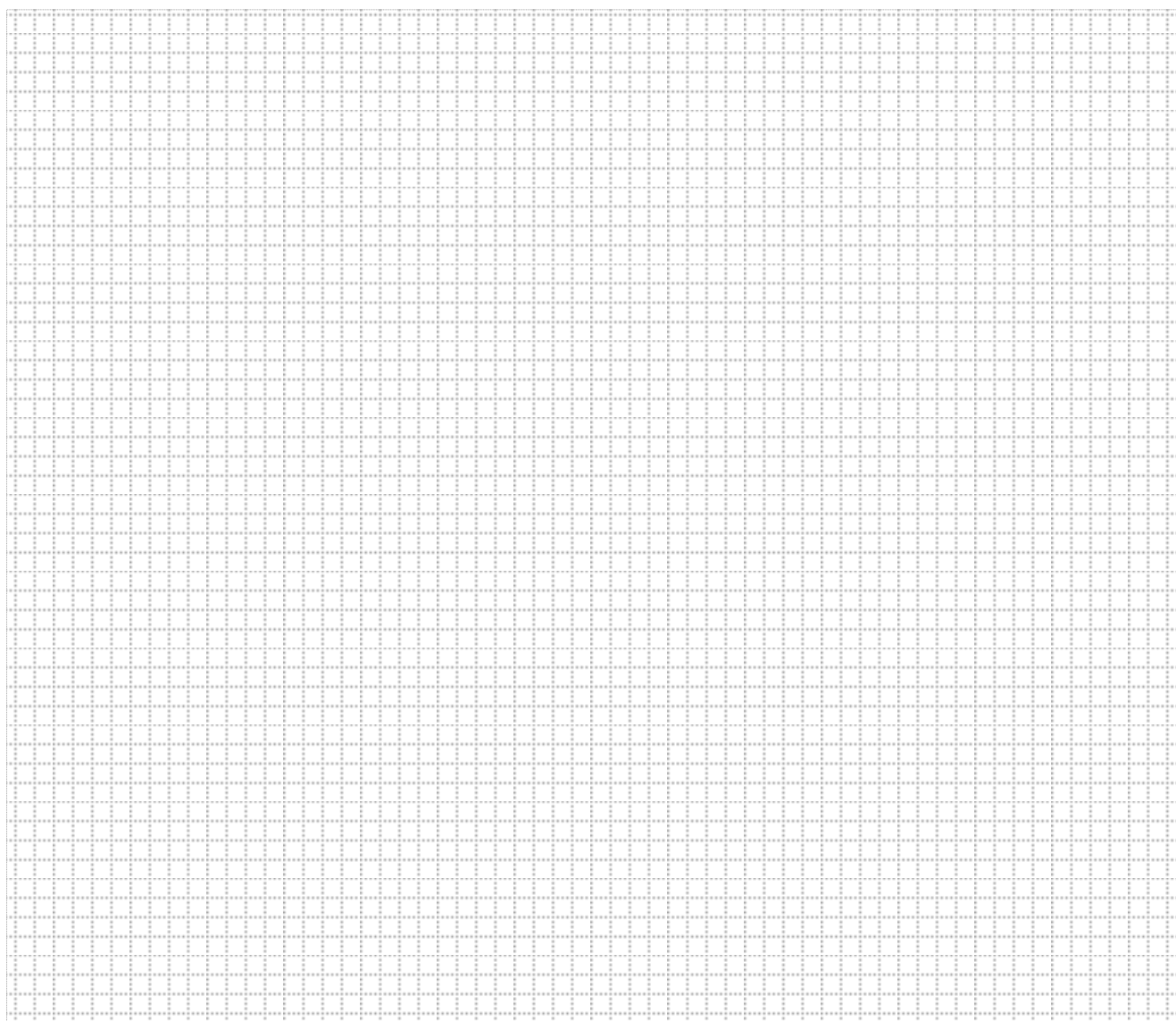


c)

Dane: n – liczba naturalna

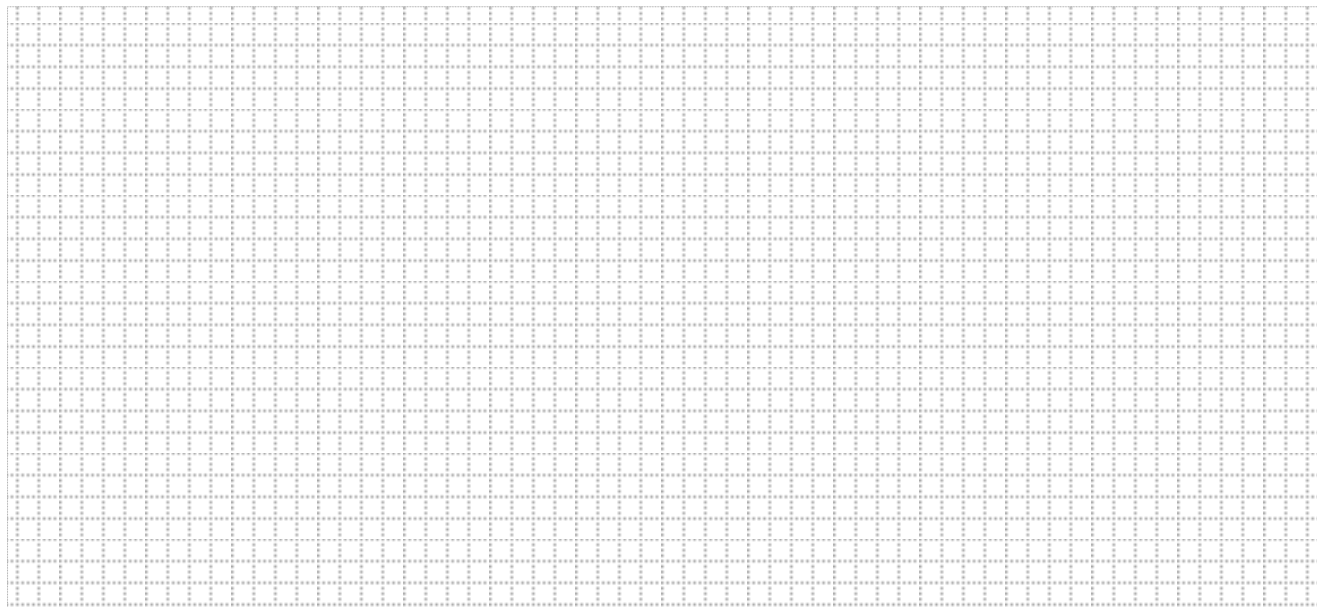
ciąg złożony z n liczb naturalnych

W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) podaj opis algorytmu, który dla danego ciągu liczb znajduje największą liczbę progów w schodach do dołu tego ciągu.



d)

Podaj, ile porównań między elementami ciągu danych w zależności od n wykonuje Twój algorytm podany w punkcie c).



Punktacja:

Wypełnia egzaminator	Podpunkt:	a)	b)	c)	d)	Razem
	Maksymalna liczba punktów:	1	2	5	1	9
	Uzyskana liczba punktów:					