

Zadanie 1. Sumy kwadratów.

Każdą dodatnią liczbę całkowitą n można reprezentować jako sumę kwadratów dodatnich liczb całkowitych. Może istnieć wiele różnych takich sum.

Przykład:

$$1 = 1^2$$

$$9 = 3^2 = 2^2 + 2^2 + 1^2 = 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2$$

Długością reprezentacji (kwadratowej) nazywamy liczbę składników sumy. W przykładzie liczba 9 ma trzy reprezentacje o długościach odpowiednio: 1, 3 i 6. Zauważ, że suma może być jednoskładnikowa.

Jedną z metod otrzymywania krótkich reprezentacji kwadratowych jest **metoda zachłanna**, w której w każdym kroku jako kolejny składnik sumy bierze się największy możliwy kwadrat liczby całkowitej gwarantujący, że suma nie przekracza n . Ta metoda nie zawsze znajdzie najkrótsze reprezentacje.

Zadanie 1.1. (0-3)

Uzupełnij poniższą tabelę zgodnie z zapisanymi w niej warunkami, czyli:

- W wierszu 3 wpisz reprezentację kwadratową liczby 23 krótszą od jej reprezentacji otrzymanej metodą zachłanną
- W wierszu 4 podaj dodatnią liczbę $n > 23$ taką, że jej reprezentacja kwadratowa otrzymana metodą zachłanną nie jest jej najkrótszą reprezentacją. Zapisz reprezentację tej liczby otrzymaną metodą zachłanną oraz reprezentację krótszą niż otrzymana metodą zachłanną.

Nr	$n > 0$	Reprezentacja kwadratowa liczby n otrzymana metodą zachłanną	Reprezentacja kwadratowa liczby n krótsza od tej otrzymanej metodą zachłanną
1	12	$3^2 + 1^2 + 1^2 + 1^2$	$2^2 + 2^2 + 2^2$
2	18	$4^2 + 1^2 + 1^2$	$3^2 + 3^2$
3	23	$4^2 + 2^2 + 1^2 + 1^2 + 1^2$	
4			

Zadanie 1.2. (0-4)

Napisz algorytm (w postaci pseudokodu lub w wybranym języku programowania), który dla danej liczby całkowitej n obliczy długość jej reprezentacji kwadratowej wyznaczonej metodą zachłanną. Twój algorytm powinien być zgodny z poniższą specyfikacją.

Uwaga: W zapisie algorytmu możesz korzystać tylko z instrukcji sterujących, operatorów arytmetycznych (dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia), operatorów logicznych, porównań i instrukcji przypisania lub **samodzielnie napisanych** funkcji i procedur. **Zabronione** jest używanie funkcji wbudowanych, dostępnych w językach programowania, zwłaszcza funkcji pierwiastek.

Specyfikacja:

Dane

n – dodatnia liczba całkowita

Wynik

dl – długość reprezentacji kwadratowej liczby n , otrzymanej metodą zachłanną

Przykład:

Dla $n = 12$ wynikiem jest $dl = 4$

A blank sheet of graph paper featuring a uniform grid of squares. The grid consists of 20 columns and 20 rows, totaling 400 small squares. The lines are thin and gray, set against a white background. There is no text or other markings on the page.

Zadanie 2. Modyfikacja tablicy.

Dane są dodatnia liczba całkowita n oraz tablica liczb całkowitych $T[1..n]$. Przeanalizuj działanie opisanej poniżej rekurencyjnej procedury $modyfikuj(s, k)$, której parametrami są dodatnie liczby całkowite s i k , $s \leq n$.

```
 $modyfikuj(s, k)$   
  Jeżeli  $s + k < n$  to  
     $modyfikuj(s + k, k)$   
   $i \leftarrow s + 1$   
  dopóki  $(i \leq n)$  oraz  $(i \leq s + k)$  wykonuj  
     $T[s] \leftarrow T[s] + T[i]$   
     $i \leftarrow i + 1$ 
```

Zadanie 2.1 (0-3)

Uzupełnij tabelę – podaj wynik działania procedury $modyfikuj$ po jej wywołaniu dla wskazanych wartości parametrów s i k .

n	Zawartość T przed wywołaniem $modyfikuj$	Wartości parametrów s i k	Zawartość T po wywołaniu $modyfikuj(s, k)$
8	[1, 1, 1, 1, 1, 1, 1, 1]	$s = 3, k = 3$	[1, 1, 6, 1, 1, 3, 1, 1]
10	[1, 4, 2, 8, 3, 6, 2, 9, 1, 5]	$s = 5, k = 6$	
13	[4, 2, 6, 2, 9, 3, 5, 2, 7, 4, 3, 2, 3]	$s = 3, k = 5$	
13	[4, 2, 6, 2, 9, 3, 5, 2, 7, 4, 3, 2, 3]	$s = 4, k = 4$	

Miejsce za obliczenia:

Zadanie 2.2 (0-2)

Dla danych n, s oraz k podaj łączną liczbę wywołań procedury `modyfikuj` dla wywołania $modyfikuj(s, k)$. Wywołanie $modyfikuj(s, k)$ jest liczone jako pierwsze.

n	s	k	łączna liczba wywołań $modyfikuj(s, k)$
5	1	3	2
2021	1	100	
2021	20	35	

Miejsce za obliczenia:

Zadanie 3 Test

Oceń prawdziwość podanych zdań. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F** – jeśli jest fałszywe.

W każdym zadaniu punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

Zadanie 3.1 (0-1)

Mamy dane operacje logiczne na bitach **not**, **and** i **or** opisane poniżej:

a	not a
1	0
0	1

a	b	a and b
1	1	1
0	1	0
1	0	0
0	0	0

a	b	a or b
1	1	1
0	1	1
1	0	1
0	0	0

oraz wyrażenie $W(a, b)$:

$$((\text{not } a) \text{ and } b) \text{ or } (a \text{ and } (\text{not } b))$$

1	$W(0,0) = 1$	P	F
2	$W(1,0) = 1$	P	F
3	$W(0,1) = 1$	P	F
4	$W(1,1) = 1$	P	F

Zadanie 3.2 (0-1)

1	$(10101)_2 + (101011)_2 = (111111)_2$	P	F
2	$(A)_{16} + (B)_{16} = (F)_{16}$	P	F
3	$(12)_8 + (12)_8 = (14)_{16}$	P	F
4	$(123)_{10} = (1111101)_2$	P	F

Zadanie 3.3 (0-1)

W pewnej bazie danych istnieją tabele: uczniowie oraz oceny połączone relacją. Tabela uczniowie składa się z kolumn: iducznia (klucz główny), imie, nazwisko, klasa, a tabela oceny składa się z kolumn: idoceny (klucz główny), iducznia (klucz obcy), ocena.

1	Zapytanie: SELECT uczniowie, klasa, Count(oceny.ocena) FROM uczniowie INNER JOIN oceny ON uczniowie, iducznia = oceny.iducznia WHERE oceny.ocena=6 GROUP BY uczniowie.klasa; da w wyniku zestawienie podające dla każdej klasy liczbę ocen celujących (6)	P	F
2	Zapytanie: SELECT Count(uczniowie. klasa) FROM uczniowie INNER JOIN oceny ON uczniowie.iducznia = oceny.iducznia WHERE oceny.ocena=6 da w wyniku zestawienie podające dla kazdej klasy liczbę ocen celujących (6)	P	F
3	Zapytanie SELECT Count(uczniowie.klasa), oceny.ocena FROM uczniowie INNER JOIN oceny ON uczniowie.iducznia = oceny.iducznia GROUP BY oceny,ocena; da w wyniku zestawienie podające dla kazdej klasy liczbę wszystkich ocen	P	F
4	Zapytanie: SELECT Count(uczniowie. klasa), oceny.ocena FROM uczniowie INNER JOIN oceny ON uczniowie.iducznia = oceny.iducznia WHERE oceny.ocena=3 GROUP BY oceny.ocena; da w wyniku zestawienie podające dla kazdej klasy liczbę ocen dostatecznych (3)	P	F