

	WYPEŁNIA ZDAJĄCY	Miejsce na naklejkę.
KOD	PESEL	Sprawdź, czy kod w lewym górnym rogu naklejki to E-100 . Jeżeli tak – przyklej naklejkę. Jeżeli nie – zgłoś to nauczycielowi.

EGZAMIN MATURALNY Z INFORMATYKI

Poziom rozszerzony Część I

DATA: 14 CZERWCA 2021 r. GODZINA ROZPOCZĘCIA: 9:00

CZAS PRACY: 60 minut

LICZBA PUNKTÓW DO UZYSKANIA: 15

WYPEŁNIA ZDAJĄCY	WYBRANE:
	(system operacyjny)
	(program użytkowy)
	(środowisko programistyczne)

Instrukcja dla zdającego

- Sprawdź, czy arkusz egzaminacyjny zawiera 8 stron (zadania 1–3).
 Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
- 2. Rozwiązania i odpowiedzi zapisz w miejscu na to przeznaczonym przy każdym zadaniu.
- 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 będą oceniane.
- 6. Wpisz zadeklarowane (wybrane) przez Ciebie na egzamin system operacyjny, program użytkowy oraz środowisko programistyczne.
- 7. Na tej stronie oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
- 8. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.



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 znajduje najkrótsze reprezentacje.

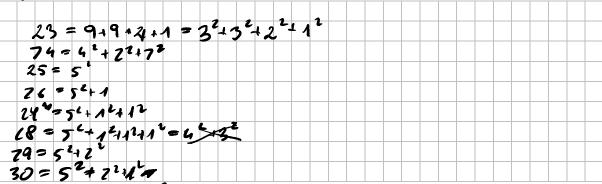
Zadanie 1.1. (0-3)

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 dodatkową 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 <i>n</i> 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	324342412	
4	32	5-42-41-41-12	42842	

Miejsce na obliczenia



31=5421+7212

Zadanie 1.2. (0-4)

Napisz algorytm (w postaci pseudokodu lub w wybranym języku programowania), który dla danej dodatniej liczby całkowitej *n* obliczy długość jej reprezentacji kwadratowej wyznaczanej 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 przypisywania lub samodzielnie napisanych funkcji i procedur wykorzystujących powyższe operacje. Zabronione jest używanie funkcji wbudowanych, dostępnych w językach programowania, zwłaszcza funkcji pierwiastek.

Specyfikacja:

Dane

n – dodatnia liczba całkowita

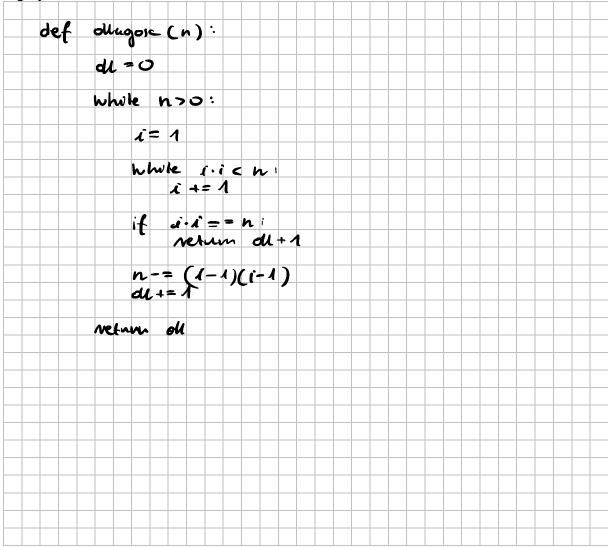
Wynik

dl – długość reprezentacji kwadratowej liczby n, otrzymanej metodą zachłanną

Przykład:

Dla n = 12 wynikiem jest dl = 4.

Algorytm:



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, $s \le n$.

modyfikuj(s, k)

jeżeli
$$s + k < n$$
 to

modyfikuj(s + k, k)

 $i \leftarrow s + 1$

dopóki ($i \le n$) oraz ($i \le s + k$) wykonuj

 $i \leftarrow i + 1$
 $i \leftarrow i + 1$

Zadanie 2.1. (0-3)

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

n	Zawartość <i>T</i> przed wywołaniem <i>modyfikuj</i>	Wartości parametrów s i <i>k</i>	Zawartość <i>T</i> po wywołaniu <i>modyfikuj</i> (s, <i>k</i>)
8	[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	1,4,2,8,74,62,9,1,5)
13	[4, 2, 6 2, 9, 3, 5, 2 7, 4, 3, 2, 3]	s = 3, k = 5	4,2,46,2,9,3,5,21,4,4,3,2,3
13	[4, 2, 6, <u>2,</u> 9, 3, 5, <u>2,</u> 7, 4, 3, <u>2,</u> 3]	s = 4, k = 4	4,2,6,40,9,3,5,21, 4,4,3,5

Miejsce na obliczenia:

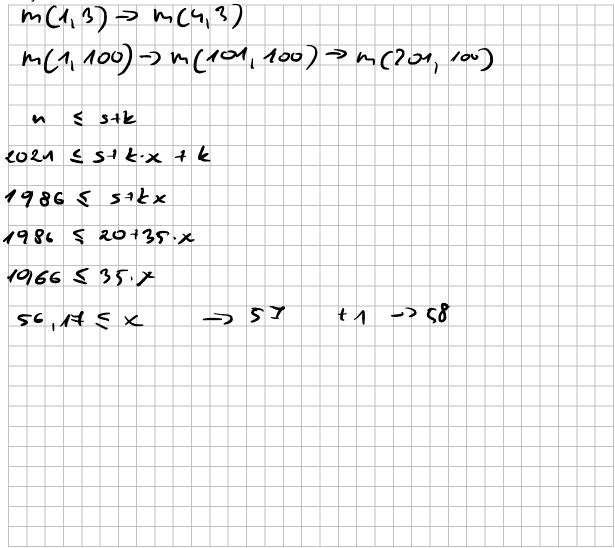
1 42 8 3 6 2 9 1 5 -> 1,4 2 8, 26, 6, 7, 9, 1 5

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 <i>modyfikuj</i> dla pierwszego wywołania <i>modyfikuj</i> (s, <i>k</i>)
5	1	3	2
2021	1	100	21
2021	20	35	58

Miejsce na obliczenia



Zadanie 3. Test

Oceń prawdziwość podanych zdań. Zaznacz \mathbf{P} , jeśli zdanie jest prawdziwe, albo \mathbf{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:

а	not a
1	0
0	1

а	b	a and b
1	1	1
0	1	0
1	0	0
0	0	0

а	b	a or b
1	1	1
0	1	1
1	0	1
0	0	0

oraz wyrażenie W(a,b): C $Q = \{a \in b : b = b \text{ or } a = b \text{ or }$

1.	W(0,0)=1	Р	(F)
2.	W(1,0)=1	(F
3.	W(0,1)=0	P /	(T)
4.	W(1,1)=0	P	F

Zadanie 3.2. (0-1)

1.	$(10101)_2 + (101011)_2 = (1111111)_2$	Р	F
2.	$(A)_{16} + (B)_{16} = (F)_{16}$	Р	F
3.	$(12)_8 + (12)_8 = (14)_{16}$	(P)	F
4.	$(123)_{10} = (1111101)_2$	Р	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	
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 każdej klasy liczbę ocen celujących (6)	P	
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 każdej 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 każdej klasy liczbę ocen dostatecznych (3)	P	F

BRUDNOPIS (nie podlega ocenie)