

WYPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

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

Miejsce na naklejkę.

Sprawdź, czy kod na naklejkę to
E-100.

Jeżeli tak – przyklej naklejkę.
Jeżeli nie – zgłoś to nauczycielowi.

EGZAMIN MATURALNY Z INFORMATYKI
POZIOM ROZSZERZONY
CZĘŚĆ I
TEST DIAGNOSTYCZNY

TERMIN: **marzec 2021 r.**

CZAS PRACY: **60 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **15**

WYPEŁNIA ZDAJĄCY

WYBRANE:

Windows 10

(system operacyjny)

MS Office 2019

(program użytkowy)

PyCharm

(środowisko programistyczne)

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 8 stron (zadania 1–3).
Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. 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.



EINP-R1-**100**-2103

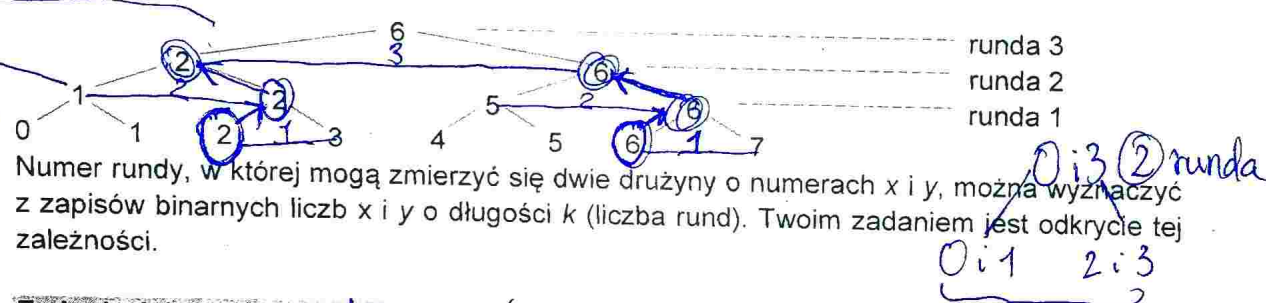
Zadanie 1. Turniej

W turnieju siatkówki bierze udział n drużyn ponumerowanych kolejnymi liczbami całkowitymi od 0 do $n-1$, gdzie $n = 2^k$ dla pewnej liczby całkowitej $k > 0$. Turniej odbywa się w rundach systemem pucharowym – przegrywający odpada z turnieju. W każdej rundzie drużyny grają w parach i do dalszej rundy przechodzi tylko zwycięzca meczu. W każdej rundzie mecze są ponumerowane kolejnymi liczbami całkowitymi, poczynając od 1. W pierwszej rundzie w meczu nr 1 grają drużyny 0 i 1, w meczu nr 2 – drużyny 2 i 3, w meczu nr 3 – drużyny 4 i 5, w meczu nr i – drużyny $2 \cdot (i-1)$ oraz $2 \cdot (i-1) + 1$, itd. W każdej z kolejnych rund w meczu nr 1 grają zwycięzcy meczów o numerach 1 i 2 z poprzedniej rundy, w meczu nr 2 – zwycięzcy meczów o numerach 3 i 4 z poprzedniej rundy, w meczu nr i – zwycięzcy meczów o numerach $2 \cdot i - 1$ oraz $2 \cdot i$ z poprzedniej rundy itd. Turniej trwa dokładnie k rund.

Przykład

Przykładową rozgrywkę w turnieju 8-drużynowym przedstawiono w postaci drzewa na rysunku poniżej. Na najniższym poziomie rysunku drzewa zapisano numery drużyn, natomiast w węzłach wewnętrznych – numery zwycięskich drużyn w poszczególnych meczach. Zwycięzcą turnieju została drużyna nr 6, która w meczu finałowym pokonała drużynę o numerze 2.

④ 0-15
 ③ 0-7
 ② 0-3
 ① 0-1
 $2^k - 1$



Numer rundy, w której mogą zmierzyć się dwie drużyny o numerach x i y , można wyznaczyć z zapisów binarnych liczb x i y o długości k (liczba rund). Twoim zadaniem jest odkrycie tej zależności.

Zadanie 1.1. (0-2) ~ 10 minut

Dla podanej liczby k (liczba rund w turnieju) oraz numerów drużyn x i y wyznacz numer rundy w turnieju, w której te dwie drużyny mogą się zmierzyć ze sobą.

k	x	y	x dwójkowo	y dwójkowo	nr rundy, w której mogą się zmierzyć drużyny x i y
3	2	6	010	110	3
4	0	3	0000	0011	2
4	30	74	0011	0111	3
5	160	304	10000	11110	4

Miejsce na obliczenia:

$$2 \cdot (i-1) \quad i \quad 2 \cdot (i-1) + 1 \rightarrow i = \text{numer meczu}$$

Zadanie 1.2. (0-4) ~ 10 minut

Napisz algorytm (w pseudokodzie lub w wybranym języku programowania), który dla danych liczb całkowitych k , x i y obliczy numer rundy w turnieju dla 2^k drużyn, w której mogą się spotkać drużyny x i y .

Uwaga: W zapisie algorytmu możesz korzystać wyłącznie z instrukcji sterujących, operatorów arytmetycznych (w tym dzielenia całkowitego i dzielenia z resztą), operatorów logicznych, porównań i instrukcji przypisywania lub samodzielnie napisanych funkcji i procedur. Zabronione jest używanie funkcji wbudowanych, dostępnych w językach programowania, a zwłaszcza funkcji podnoszącej do potęgi.

Specyfikacja algorytmu**Dane**

k – dodatnia liczba całkowita, liczba rund w turnieju

x , y – dwie różne liczby całkowite z przedziału $[0, 2^k - 1]$, numery drużyn

Wynik

runda – nr rundy, w której mogą się spotkać drużyny x i y

Algorytm:

```
jeśli  $x < y$ :  
     $max = y - x$   
w innym wypadku:  
     $max = x - y$   
 $counter = 1$   
 $num = 2$   
wykonuj  $k$  razy:  
    jeśli  $num - 1 \geq max$ :  
        wypisz wartość  $counter$  i zakończ program  
    w innym wypadku:  
         $num = num \cdot 2$   
         $counter = counter + 1$ 
```

Wypełnia egzaminator	Nr zadania	2.1.	2.2.
	Maks. liczba pkt.	2	4
	Uzyskana liczba pkt.		

Zadanie 2. Analiza algorytmu

Wykonaj analizę funkcji $Algo(n)$, której argumentem jest dodatnia liczba całkowita n .

$Algo(n)$

jeżeli $n \leq 2$ to

wynikiem jest 1

w przeciwnym przypadku

$p \leftarrow 1$

$k \leftarrow n$

→ dopóki $k - p > 1$ wykonuj

$s \leftarrow (p + k) \text{ div } 2$

jeżeli $s * s \leq n$ to

$p \leftarrow s$

w przeciwnym przypadku

$k \leftarrow s$

wynikiem jest p

def $Algo(n)$:
if $n \leq 2$:
return 1

else:

$p = 1$

$k = n$

while $k - p > 1$:

$s = (p + k) // 2$

if $s * s \leq n$:

$p = s$

else:

$k = s$

Uwaga: div oznacza dzielenie całkowite.

Zadanie 2.1. (0-2)

~ 7 minut

Uzupełnij tabelę – podaj wynik funkcji $Algo$ dla podanych w tabeli wartości n .

n	Wynik otrzymany po wywołaniu $Algo(n)$
5	2
35	5
1025	32

Miejsce na obliczenia:

	p	k	k-p	p+k	s ²	s																																	
n=5	1	5	4	6	9	3	{k-p=1 ≤ 1	1026	513+1																														
	1	3	2	4	4	2		257+1																															
n=35	1	35	34	36	18 ²	18	p=5	128	64																														
	1	18	17	19	9 ²	9		32	?																														
	1	9	8	10	5 ²	5																																	
n=1025	1	1025	1024	1026	513 ²	513	<table><tr><th>p</th><th>k</th><th>k-p</th><th>p+k</th><th>s²</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	p	k	k-p	p+k	s ²																											
p	k	k-p	p+k	s ²																																			
	1	513	512	514	257 ²	257																																	
	1	257	256	258	129 ²	129																																	
	1	129	128	130	75 ²	75																																	
→	1	75	74	76	38 ²	38																																	
	38	75	37	113	56 ²	56																																	

Strona 4 z 8

EINP-R1_100

Zadanie 2.2. (0-3) *~ 3 minuty*

Uzupełnij tabelę – podaj liczbę wykonań instrukcji „ $s \leftarrow (p + k) \div 2$ ” podczas obliczania wartości funkcji $Algo(n)$ dla podanych wartości n .

n	Liczba wykonań instrukcji „ $s \leftarrow (p + k) \div 2$ ” podczas obliczania wartości funkcji $Algo(n)$
5	2
2	0
63	3
1024	5

Miejsce na obliczenia

$63 \rightarrow 1$
 $31 \rightarrow 2$
 $15 \rightarrow 2 \rightarrow 63$
 $7 \leq 63$
 3
 1

1024
 512
 256
 128
 64
 $32^2 = 1024$

$$63+1=64$$

$$64$$

$$32+1$$

$$16+1 \rightsquigarrow 8$$

$$8+1 \rightsquigarrow 4$$

$$4+1$$

Wypełnia egzaminator	Nr zadania	2.1.	2.2.
	Maks. liczba pkt.	2	3
	Uzyskana liczba pkt.		

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)

~ 2 min

W komórce C1 arkusza kalkulacyjnego zapisano formułę:

=JEŻELI(ORAZ(MOD(A1;2)=1;MOD(B1;2)=1);A1+B1;A1*B1)

niepar i niepar

inne kombinacje

1.	Jeśli w A1 wpisano liczbę <u>1</u> , a w B1 liczbę <u>3</u> , to w C1 w wyniku obliczenia formuły pojawi się liczba 4.	P	F
2.	Jeśli w A1 wpisano liczbę <u>4</u> , a w B1 liczbę <u>3</u> , to w C1 w wyniku obliczenia formuły pojawi się liczba 3.	P	F
3.	Jeśli w A1 i B1 wpiszemy <u>dowolną liczbę całkowitą dodatnią</u> , to w wyniku obliczenia formuły w C1 zawsze pojawi się liczba parzysta.	P	F
4.	Jeśli w A1 i B1 wpiszemy <u>dowolną liczbę całkowitą dodatnią</u> , to w wyniku obliczenia formuły w C1 zawsze pojawi się liczba większa niż 1.	P	F

$1+3 \rightarrow 4$
 $2 \cdot 3 \rightarrow 6$
 $2 \cdot 2 \rightarrow 4$
 $1+1=2$
 $2 \cdot 1=2$

Zadanie 3.2. (0-1)

~ 3 min

Mamy dane operacje (bramki) logiczne na bitach: not oraz and 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

$$\sim(\sim p \wedge q) \wedge \sim(p \wedge \sim q)$$

$$\Leftrightarrow (p \vee \sim q) \wedge (\sim p \vee q)$$

oraz wyrażenie $W(a,b)$:

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

1.	$W(0,0)=0$	$(0 \vee 1) \wedge (1 \vee 0) \Leftrightarrow 1$	P	F
2.	$W(1,0)=0$	$(1 \vee 1) \wedge (0 \vee 0) \Leftrightarrow 0$	P	F
3.	$W(0,1)=1$	$(0 \vee 0) \wedge (0 \vee 1) \Leftrightarrow 0$	P	F
4.	$W(1,1)=1$	$(1 \vee 0) \wedge (0 \vee 1) \Leftrightarrow 1$	P	F

$$\begin{array}{r} 1011101 \\ - 10111 \\ \hline \end{array}$$

Zadanie 3.3. (0-1) ~ 2 min 1000110 → 7 cyfr
Różnica $1011101_2 - 10111_2$ dwóch liczb zapisanych w systemie binarnym jest:

1.	mniejsza niż $100111_2 \rightarrow 6$ cyfr	P	F
2.	równa 1000110_2	P	F
3.	większa niż $10111_2 \rightarrow 5$ cyfr	P	F
4.	równa 1001000_2	P	F

Zadanie 3.4. (0-1) ~ 2 min
W bazie danych istnieje tabela *oceny*(*id_oceny*, *id_ucznia*, *przedmiot*, *ocena*), zawierająca następujące dane:

id_oceny	id_ucznia	przedmiot	ocena
1	1	matematyka	3
2	1	informatyka	4
3	1	fizyka	2
4	2	matematyka	6
5	2	fizyka	3
6	2	informatyka	5
7	3	matematyka	4
8	3	fizyka	2
9	3	informatyka	3

1.	Wynikiem zapytania <u>SELECT COUNT(id_ucznia) FROM oceny;</u> jest 3	P	F
2.	Wynikiem zapytania <u>SELECT COUNT(id_ucznia) FROM oceny</u> <u>WHERE przedmiot="fizyka";</u> jest 3	P	F
3.	Wynikiem zapytania <u>SELECT COUNT(przedmiot) FROM oceny;</u> jest 9 → ok, liczy się każdy	P	F
4.	Wynikiem zapytania <u>SELECT COUNT(przedmiot) FROM oceny</u> <u>(WHERE ocena > 3;</u> jest 4	P	F

brak grupowania, liczy się każdy
↓
9

Wypełnia egzaminator	Nr zadania	3.1.	3.2.	3.3.	3.4.
	Maks. liczba pkt.	1	1	1	1
	Uzyskana liczba pkt.				