

Arkusz zawiera informacje prawnie chronione do momentu rozpoczęcia egzaminu.

Układ graficzny © CKE 2013

UZUPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

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

*Miejsce
na naklejkę
z kodem*

**EGZAMIN MATURALNY
Z INFORMATYKI**

POZIOM ROZSZERZONY

CZĘŚĆ I

19 MAJA 2015

**Godzina rozpoczęcia:
9:00**

WYBRANE:

.....
(środowisko)
.....
(kompilator)
.....
(program użytkowy)

**Czas pracy:
90 minut**

**Liczba punktów
do uzyskania: 20**

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 11 stron. 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. Na tej stronie oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
9. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.



MIN-R1_1P-152

Zadanie 1. Potęgi dwójki

Każdą liczbę całkowitą dodatnią n możemy zapisać jako sumę różnych potęg liczby 2.

Przykład:

$$11 = 1 + 2 + 8 = 2^0 + 2^1 + 2^3$$

$$18 = 2 + 16 = 2^1 + 2^4$$

Zadanie 1.1. (1 pkt)

Podane w tabeli liczby n zapisz jako sumę różnych potęg liczby 2.

n	składniki sumy
11	$1 + 2 + 8$
23	
50	

Miejsce na obliczenia.

Zadanie 1.2. (5 pkt)

Zapisz algorytm (w postaci listy kroków, schematu blokowego lub w wybranym języku programowania) obliczający liczbę różnych potęg liczby 2, których suma jest równa danej liczbie n . Twój algorytm powinien być zgodny z poniższą specyfikacją.

Uwaga: W zapisie możesz wykorzystać operacje dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i dzielenia z resztą lub samodzielnie napisane funkcje.

Specyfikacja algorytmu:

Dane:

n – dodatnia liczba całkowita

Wynik:

lp – liczba różnych potęg liczby 2, których suma daje liczbę n

Przykład:

Dla $n = 18$ wynikiem jest $lp = 2$.

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin, dark gray lines. There are no margins, text, or other markings on the page.

Wypełnia egzaminator	Nr zadania	1.1.	1.2.
	Maks. liczba pkt.	1	5
	Uzyskana liczba pkt.		

Zadanie 2. Koszyk zabawek

Wyobraź sobie, że w pewnym sklepie z zabawkami wygrałeś „koszyk zakupów”, którego zawartość nie może łącznie ważyć więcej niż 10 kg. Oto artykuły, z których możesz wybierać:

Lp.	nazwa artykułu	masa	cena	$\frac{\text{cena}}{\text{masa}}$
1.	rowerek	8 kg	320 zł	40
2.	wózek dla lalek	4 kg	152 zł	38
3.	lalka	1 kg	37 zł	37
4.	duży miś	2 kg	70 zł	35
5.	klocki	3 kg	99 zł	33
6.	hulajnoga	5 kg	155 zł	31
7.	mały miś	1 kg	30 zł	30

Ponieważ wszystkie zabawki są dla Ciebie tak samo atrakcyjne, chcesz wybrać zabawki do koszyka tak, żeby ich łączna wartość była jak największa. Przy podejmowaniu decyzji o wyborze zabawek możesz skorzystać z jednej z trzech strategii:

- I. Wybierasz zabawki od najdroższej do najtańszej, kontrolując jednocześnie masę zabawek w koszyku, żeby nie przekroczyć ograniczenia na łączną masę jego zawartości. W przypadku takiej samej ceny wybierasz zabawkę lżejszą.
- II. Wybierasz zabawki od najlżejszej do najcięższej, kontrolując jednocześnie masę zabawek w koszyku. W przypadku takiej samej masy zabawek wybierasz zabawkę droższą.
- III. Wybierasz zabawki w kolejności od największego do najmniejszego ilorazu ceny do masy $\left(\frac{CENA[i]}{MASA[i]}\right)$, kontrolując jednocześnie masę zabawek w koszyku.

Jeżeli więcej niż jedna zabawka spełnia kryterium wyboru, to wybierasz dowolną z takich zabawek.

Zadanie 2.1. (1 pkt)

Jaka będzie zawartość koszyka przy zastosowaniu każdej ze strategii: I, II, III, i przy założeniu, że te same zabawki możemy do koszyka wybierać **wielokrotnie**, o ile tylko nie przekroczymy dozwolonej, całkowitej masy zakupów? Uzupełnij tabelę: podaj nazwy wybranych zabawek, liczby ich egzemplarzy oraz sumaryczną wartość zabawek w koszyku.

	Strategia I	Strategia II	Strategia III
Zawartość koszyka			
Wartość koszyka w zł			

Miejsce na obliczenia.

Zadanie 2.2. (2 pkt)

Uzupełnij poniższy algorytm, który oblicza wartość koszyka przy wyborze zabawek zgodnym ze strategią III. Artykuły w koszyku **mogą się powtarzać**. W algorytmie wykorzystano strategię III uwzględniającą równocześnie masy artykułów i ich ceny.

Specyfikacja:

Dane:

mk – ograniczenie na łączną masę zawartości koszyka

n – liczba dostępnych artykułów

$MASA[1..n]$ – tablica n -elementowa zawierająca masy *dostępnych zabawek*

w kolejności nierosnących ilorazów ceny do masy $\left(\frac{CENA[i]}{MASA[i]} \geq \frac{CENA[i+1]}{MASA[i+1]}\right)$

$CENA[1..n]$ – tablica n -elementowa zawierająca ceny *dostępnych zabawek*

w kolejności nierosnących ilorazów ceny do masy $\left(\frac{CENA[i]}{MASA[i]} \geq \frac{CENA[i+1]}{MASA[i+1]}\right)$

mk , n oraz ceny i masy są dodatnimi liczbami całkowitymi.

Wynik:

$K[1..n]$ – tablica n -elementowa liczb całkowitych, gdzie $K[i]$ jest liczbą egzemplarzy i -tej zabawki zapakowanej do koszyka zgodnie ze strategią III

w – łączna wartość zabawek w koszyku

krok 1: Dla $i = 1$ do n wykonaj $K[i] \leftarrow 0$

krok 2:

krok 3: $i \leftarrow 1$

krok 4: Dopóki $i \leq n$ oraz $mk > 0$

krok 5: $K[i] \leftarrow mk \text{ div } MASA[i]$

krok 6: $mk \leftarrow mk \text{ mod } MASA[i]$

krok 7:

krok 8: $i \leftarrow i + 1$

Uwaga:

Operatory mod i div oznaczają – odpowiednio – resztę z dzielenia i dzielenie całkowite.

Wypełnia egzaminator	Nr zadania	2.1.	2.2.
	Maks. liczba pkt.	1	2
	Uzyskana liczba pkt.		

Zadanie 2.3. (1 pkt)

Jaka będzie zawartość koszyka przy zastosowaniu każdej ze strategii: I, II, III, i przy założeniu, że zabawki **nie mogą się powtarzać**? Uzupełnij tabelę: podaj nazwy wybranych zabawek i sumaryczną wartość koszyka.

	Strategia I	Strategia II	Strategia III
Zawartość koszyka			
Wartość koszyka w zł			

Miejsce na obliczenia.

Zadanie 2.4. (4 pkt)

Zaprojektuj i zapisz (w postaci listy kroków, schematu blokowego lub kodu wybranego języka programowania) algorytm stosujący strategię III dobierania zabawek do koszyka tak, aby wybrane zabawki w koszyku **nie mogły się powtarzać**.

Specyfikacja:

Dane:

mk – ograniczenie na łączną masę zawartości koszyka

n – liczba dostępnych artykułów

$MASA[1..n]$ – tablica n -elementowa zawierająca masy *dostępnych zabawek*
w kolejności nierosnących ilorazów ceny do masy $\left(\frac{CENA[i]}{MASA[i]} \geq \frac{CENA[i+1]}{MASA[i+1]}\right)$

$CENA[1..n]$ – tablica n -elementowa zawierająca ceny *dostępnych zabawek* w
kolejności nierosnących ilorazów ceny do masy $\left(\frac{CENA[i]}{MASA[i]} \geq \frac{CENA[i+1]}{MASA[i+1]}\right)$

mk , n oraz ceny i masy są dodatnimi liczbami całkowitymi.

Wynik:

$K[1..n]$ – tablica n -elementowa, gdzie $K[i]$ jest równe 1, gdy i -ta zabawka została dodana do koszyka, a 0 w przeciwnym wypadku.

w – łączna wartość zabawek w koszyku

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin, dark gray lines. The grid covers the entire area of the page, leaving no margins or other markings. There are 20 columns and 20 rows of squares, creating a total of 400 square units.

Wypełnia egzaminator	Nr zadania	2.3.	2.4.
	Maks. liczba pkt.	1	4
	Uzyskana liczba pkt.		

Zadanie 3. Test

Oceń, czy poniższe informacje są prawdziwe. Zaznacz P, jeśli informacja jest prawdziwa, albo F, jeśli jest fałszywa. W każdym zadaniu punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

Zadanie 3.1. (1 pkt)

Dla danej tablicy *Tab*

	<i>Tab</i>	<i>k</i>			
		1	2	3	4
<i>w</i>	1	1	1	4	4
	2	2	2	3	3
	3	3	3	2	2
	4	4	4	1	1

wykonano poniższy algorytm:

$k \leftarrow 1; s \leftarrow 0; w \leftarrow 0;$

powtarzaj

$w \leftarrow w + 1;$

$s \leftarrow s + Tab[w, k];$

aż $w = 4;$

wypisz $s, w, k;$

W wyniku zostaną wypisane liczby:

1.	10, 4, 1	P	F
2.	10, 1, 1	P	F
3.	40, 4, 4	P	F
4.	40, 4, 1	P	F

Zadanie 3.2. (1 pkt)

Poniżej zapisano wyrażenia w odwrotnej notacji polskiej (ONP). Wartościami tych wyrażeń są:

	Wyrażenie ONP	Wartość wyrażenia		
1.	7 3 - 2 /	2	P	F
2.	4 3 - 1 3 + *	8	P	F
3.	3 5 1 - *	12	P	F
4.	8 2 + 2 /	10	P	F

Zadanie 3.3. (1 pkt)

Pewna podsieć ma maskę: 255.255.255.248. Ile maksymalnie komputerów można podłączyć do danej podsieci? Uwzględnij, że 2 z możliwych adresów w sieci to adres sieci oraz adres rozgłoszeniowy.

1.	10	P	F
2.	8	P	F
3.	6	P	F
4.	4	P	F

Zadanie 3.4. (1 pkt)

W celu posortowania rosnąco ciągu liczb [2, 1, 0, 3] wykonano porównania i ewentualnie zamieniono liczby w parach otoczonych owalami. Jakie to sortowanie?

2 1 0 3
1 2 0 3
1 0 2 3

1 0 2 3
0 1 2 3

0 1 2 3

0 1 2 3

1.	przez wstawianie	P	F
2.	bąbelkowe	P	F
3.	kubelkowe	P	F
4.	szybkie	P	F

Zadanie 3.5. (1 pkt)

Program rozpowszechniany za darmo, z którego możemy korzystać w pełni przez nieograniczony czas, może być na licencji

1.	shareware.	P	F
2.	freeware.	P	F
3.	adware.	P	F
4.	GNU GPL.	P	F

Wypełnia egzaminator	Nr zadania	3.1.	3.2.	3.3.	3.4.	3.5.
	Maks. liczba pkt.	1	1	1	1	1
	Uzyskana liczba pkt.					

Zadanie 3.6. (1 pkt)

Dana jest tabela *Lista*:

Lp.	Imie	Nazwisko	Miasto	DataUrodzenia
1.	Marian	Kubok	Gdynia	1980-12-07
2.	Michalina	Przybysz	Kraków	1995-06-06
3.	Marcelina	Marchewka	Mikołajki	1988-02-09
4.	Zygmunt	Piotrowski	Katowice	1999-04-15

Zastosowanie dla powyższej tabeli *Lista* zapytania

SELECT *Imie*, *Nazwisko*

FROM *Lista*

WHERE Year(*DataUrodzenia*)>1990

ORDER BY *Nazwisko*;

spowoduje wypisanie:

1.	Imie	Nazwisko		P	F
	Zygmunt	Piotrowski			
	Michalina	Przybysz			
2.	Imie	Nazwisko		P	F
	Michalina	Przybysz			
	Zygmunt	Piotrowski			
3.	Imie	Nazwisko		P	F
	Marcelina	Marchewka			
	Marian	Kubok			
4.	Imie	Nazwisko		P	F
	Marian	Kubok			
	Marcelina	Marchewka			

Wypełnia egzaminator	Nr zadania	3.6.
	Maks. liczba pkt.	1
	Uzyskana liczba pkt.	

BRUDNOPIS (*nie podlega ocenie*)