
PRÓBNY EGZAMIN MATURALNY Z INFORMATYKI

POZIOM ROZSZERZONY Część I

Czas pracy: **60 minut**

Liczba punktów do uzyskania: **15**

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 10 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 poniżej 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, pseudokodu lub języka programowania, który wybrałaś/eś na egzamin.
-

Dane uzupełnia uczeń:

WYBRANE:

.....
(środowisko)

.....
(kompilator)

.....
(program użytkowy)

PESEL:

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

Klasa:

--	--	--

ZADANIE 1. TEST (4 PUNKTY)

Oceń, czy poniższe zdania są prawdziwe. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F**, jeśli zdanie jest fałszywe. W każdym zadaniu cząstkowym punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

ZADANIE 1.1 (0-1)

Dla zmniejszenia liczby mnożeń przy obliczaniu wartości dziesiętnej liczby binarnej wykorzystuje się:

1.	algorytm Huffmana	P	F
2.	szyfr Cezara	P	F
3.	schemat Hornera	P	F
4.	ciąg Fibonacciego	P	F

ZADANIE 1.2 (0-1)

Dla wyrażeń arytmetycznych: $2*4 + (3 - 7)$ oraz $2*(4+ (3 - 7))$ prawdziwe są następujące zdania:

1.	Obydwa wyrażenia mają tę samą wartość liczbową.	P	F
2.	Obydwa wyrażenia mają identyczną postać po ich zapisaniu w ONP (Odwrotnej Notacji Polskiej).	P	F
3.	Notacja ONP dla pierwszego wyrażenia ma postać: $2\ 4\ *\ 3\ 7\ -\ +$	P	F
4.	Notacja ONP dla drugiego wyrażenia ma postać: $2\ 4\ *\ 3\ 7\ -\ +$	P	F

ZADANIE 1.3 (0-1)

W grafice wektorowej:

1.	Przechowywana jest informacja o kształtach i ich położeniu.	P	F
2.	Przechowywana jest informacja o poszczególnych pikselach.	P	F
3.	Zmiana rozmiaru obrazu powoduje pogorszenie jego jakości.	P	F
4.	Rysunki zachowujemy w plikach o formacie png.	P	F

ZADANIE 1.4 (0-1)

Za tłumaczenie adresów domenowych na adresy IP odpowiada serwer:

1.	DHCP	P	F
2.	SMTP	P	F
3.	DNS	P	F
4.	FTP	P	F

Wypełnia egzaminator	Numer zadania	1.1	1.2	1.3	1.4
	Maksymalna liczba punktów	1	1	1	1
	Uzyskana liczba punktów				

ZADANIE 2. WYDAWANIE RESZTY (0-6)

Jednym ze znanych problemów informatycznych jest **problem wydawania reszty**. Mając daną kwotę K i zestaw dostępnych nominałów banknotów i monet, należy znaleźć najmniejszą liczbę banknotów i monet potrzebną do wydania kwoty równej K . Poniżej podano specyfikację tego problemu.

Specyfikacja:

Dane:

K - liczba naturalna oznaczająca kwotę do wydania

N - liczba naturalna oznaczająca liczbę dostępnych nominałów

dostępne nominały banknotów i monet zapamiętane w tablicy $Nomin[]$, o rozmiarze N .

Wynik:

Liczba naturalna równa najmniejszej liczbie banknotów i monet, za pomocą których można wydać kwotę równą K .

ZADANIE 2.1 (0-1)

W zadaniu poniżej przyjmijmy, że mamy dostępne następujące polskie nominały wyrażone w złotych: $Nomin = [1, 2, 5, 10, 20, 50, 100, 200]$.

Użyj tych nominałów do wydania kwot określonych w tabeli. Wykonaj obliczenia i uzupełnij puste pola tabeli.

Kwota	Najmniejsza liczba banknotów i monet	Użyte banknoty i monety
47	4	20, 20, 5, 2
84		
533		

ZADANIE 2.3 (0-2)

Algorytm zachłanny nie zawsze oblicza najlepsze rozwiązanie. Przyjmijmy następującą tablicę nominałów: $Nomin = [1, 2, 7, 10]$. Dla kwoty $K=35$ zachłanne rozwiązanie obliczy liczbę banknotów i monet równą 6 (10, 10, 10, 2, 2, 1), podczas gdy najmniejsza liczba banknotów i monet jest równa 5 (7, 7, 7, 7, 7 lub 10, 10, 7, 7, 1).

Poprawne rozwiązanie możemy obliczyć za pomocą algorytmu dynamicznego. Algorytm polega na przetwarzaniu kolejnych nominałów oraz obliczaniu najmniejszej liczby banknotów i monet potrzebnych do wydania kwot równych kolejno od 0 do K . Przy obliczaniu wyniku dla kolejnego nominału wykorzystywane są informacje pozyskane w czasie wcześniejszych obliczeń. Jeśli nie będzie możliwe wydanie kwoty przy użyciu dostępnych nominałów, zostanie zwrócony wynik „nieskończoność”.

Posługując się poniżej opisanym algorytmem dynamicznym, uzupełnij tablicę P w pustych miejscach wartościami obliczanymi w kolejnych iteracjach pętli z kroku 4.

1. Utwórz tablicę P o rozmiarze $K+1$
2. Wypełnij tablicę P wartościami równymi „nieskończoność”.
3. $P[0] := 0$
4. Dla kolejnego nominału **nom** z tablicy $Nomin$, wykonuj:
 5. Od $i:=0$ do $K-nom$, wykonuj:
 6. Jeżeli $P[i+1] < P[i+nom]$, to:
 7. $P[i+nom] := P[i]+1$
 8. Wypisz $P[K]$

Kolejne postaci tablicy P .

indeksy P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
początkowe wartości P	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

indeksy P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
wartości P dla nominału 1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

BRUDNOPIS (*nie podlega ocenie*)