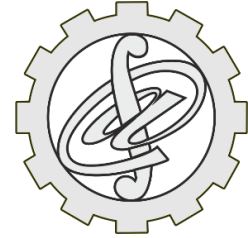




## POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej  
Studenckie Koło Naukowo-Informatyczne „Link”  
ul. Kaszubska 23, 44-100 Gliwice



### ZADANIE 1 – ZADANIE DO PO(D)CIĄGU

Rozpatrzmy ciąg o danej długości, którego elementami są zera lub jedynek. Część z tych ciągów ma taką własność, że możemy z niego wybrać taki podciąg (zachowując kolejność elementów), który jest taki sam jak podciąg złożony z pozostałych elementów tego ciągu. Przykładowo, ciąg  $\{1,1,0,0,0,1,0,1,1,1\}$  ma taką cechę, bo można z niego wybrać odpowiedni podciąg (kolor czerwony), który pozostawi elementy danego ciągu (kolor niebieski) tworzące ten sam podciąg (takich wyborów może być więcej):

$$\{1,1,0,0,0,1,0,1,1,1\}.$$

Istnieją też ciągi, dla których nie można wybrać takiego podciągu, np.  $\{1,0,0,1\}$ .

Napisz program, który dla zadanej liczby naturalnej  $n > 1$  rozpatrywał będzie wszystkie ciągi długości  $n$  zer i jedynek i zliczał będzie, ile wśród nich jest ciągów o podanej cesze.

Przykładowo, dla  $n = 4$  otrzymalibyśmy ciągi:

$$\begin{array}{cccc} \{0,0,0,0\}, & \{0,0,0,1\}, & \{0,0,1,0\}, & \{0,0,1,1\}, \\ \{0,1,0,0\}, & \{0,1,0,1\}, & \{0,1,1,0\}, & \{0,1,1,1\}, \\ \{1,0,0,0\}, & \{1,0,0,1\}, & \{1,0,1,0\}, & \{1,0,1,1\}, \\ \{1,1,0,0\}, & \{1,1,0,1\}, & \{1,1,1,0\}, & \{1,1,1,1\}, \end{array}$$

z których 6 ma tę cechę (znaczenie kolorów jak w poprzednim przykładzie).

Dla  $n = 3$  żaden z ciągów:

$$\{0,0,0\}, \quad \{0,0,1\}, \quad \{0,1,0\}, \quad \{0,1,1\}, \quad \{1,0,0\}, \quad \{1,0,1\}, \quad \{1,1,1\}$$

nie ma takiej cechy – program zwróciłby wartość zero.

Dla większych wartości  $n$  liczba wykonanych obliczeń jest znacząca. Postaraj się zwiększyć wydajność swojego programu. Sprawdzając to zadanie komisja przyzna pięciu autorom dodatkowe punkty, które przypisane będą pięciu najszybciej działającym programom.

## ALGORYTMION

Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



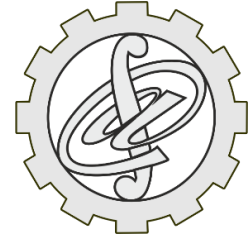
Studenckie Koło Naukowo-  
Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



## POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej  
Studenckie Koło Naukowo-Informatyczne „Link”  
ul. Kaszubska 23, 44-100 Gliwice



### ZADANIE 2 – ROZKŁAD (ALE NIE ZEPSUCIE)

Dowolną liczbę naturalną  $n > 3$  można przedstawić jako sumę liczb pierwszych. Inaczej niż w przypadku iloczynu, rozkład taki nie jest jednoznaczny (w ogólnym przypadku). Przykładowo, dla  $n = 5$  mamy:

$$5 = 2 + 3,$$

a dla  $n = 10$  otrzymamy następujące rozkłady:

$$10 = 2 + 2 + 2 + 2 + 2,$$

$$10 = 2 + 2 + 3 + 3,$$

$$10 = 2 + 3 + 5,$$

$$10 = 5 + 5,$$

$$10 = 3 + 7.$$

Założmy, że przestawianie składników nie zmienia rozkładu, tzn. że np. rozkłady  $10 = 3 + 7$  i  $10 = 7 + 3$ , to są te same rozkłady.

Zadanie to jest matematyczno-informatyczne – punkty otrzymać można za część matematyczną, w której należy wykazać słuszność twierdzenia:

Każdą liczbę naturalną  $n > 3$  można na co najmniej jeden sposób przedstawić jako sumę liczb pierwszych.

oraz za część informatyczną, w której należy napisać program, który dla zadanej liczby naturalnej  $n > 3$  wypisywał będzie (forma tego zapisu pozostaje w gestii rozwiązującego) wszystkie możliwe rozkłady tej liczby na sumę liczb pierwszych z dokładnością do kolejności składników (kolejność sum również nie ma znaczenia) oraz liczbę tych rozkładów.

Przykładowo, podając  $n = 5$  lub  $n = 10$  program wypisałby przedstawione powyżej rozkłady i dodatkowo informację, że rozkładów tych jest 1 (dla  $n = 5$ ) lub 5 (dla  $n = 10$ ).

## ALGORYTMION

Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



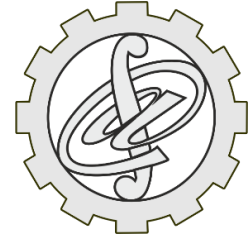
Studenckie Koło Naukowo-  
Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



## POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej  
Studenckie Koło Naukowo-Informatyczne „Link”  
ul. Kaszubska 23, 44-100 Gliwice



### ZADANIE 3 – ZADANIE GRUPOWE

W matematyce ważnym pojęciem jest pojęcie *grupy*.

*Grupą nazywamy zbiór  $G$  wraz z działaniem<sup>1</sup>  $\star$  wtedy i tylko wtedy, gdy:*

*a) działanie  $\star$  jest łączne, tzn. dla wszystkich  $a, b, c \in G$  mamy:*

$$(a \star b) \star c = a \star (b \star c);^2$$

*b) w zbiorze  $G$  istnieje element  $e$  (element neutralny), taki że dla dowolnego elementu  $a$  zbioru  $G$  mamy:*

$$a \star e = e \star a = a;$$

*c) w zbiorze  $G$  każdy element  $a$  ma element odwrotny  $a^{-1}$ , taki że:*

$$a \star a^{-1} = a^{-1} \star a = e.$$

*Jeżeli dodatkowo działanie  $\star$  jest przemienne, tzn. dla dowolnej pary elementów zbioru  $G$  mamy:*

$$a \star b = b \star a,$$

*to grupę taką nazywamy grupą przemienną (abelową).*

Zbiory  $G$ , dla których określamy działanie  $\star$  mogą być skończone lub nieskończone. My zajmować będziemy się zbiorami skończonymi (w części informatycznej tego zadania) jak też nieskończonymi (w części matematycznej).

<sup>1</sup> Działanie dwuargumentowe w zbiorze  $G$  oznacza, że dla dowolnej pary elementów tego zbioru wynik tego działania również jest w tym zbiorze. Przykładowo dodawanie w zbiorze liczb całkowitych jest działaniem (suma dwóch dowolnych liczb całkowitych jest liczbą całkowitą), ale np. funkcja  $\odot$  określona następującą zależnością:  $\odot(x, y) = \frac{x}{y^2+1}$  nie jest działaniem w zbiorze liczb całkowitych, bo np.  $\odot(7, -3) = \frac{7}{(-3)^2+1} = \frac{7}{10}$ , a wynik ten nie jest liczbą całkowitą.

<sup>2</sup> Oznaczenie  $a \star b$  rozumiemy jako  $\star(a, b)$ .

**ALGORYTMION**

Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



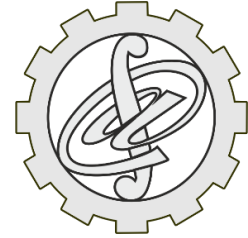
Studenckie Koło Naukowo-  
Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



# POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej  
Studenckie Koło Naukowo-Informatyczne „Link”  
ul. Kaszubska 23, 44-100 Gliwice



Podobnie jak zadanie 2, zadanie to jest matematyczno-informatyczne – punkty otrzymać można za część matematyczną, w której należy sprawdzić, który ze zbiorów ( $\mathbb{Z}$  oznacza zbiór liczb całkowitych, a  $\mathbb{Q}$  zbiór liczb wymiernych) wraz z podanym działaniem jest grupą (grupą abelową) – odpowiedź uzasadnij:

1.  $G = \mathbb{Q}, a * b = \frac{a+b}{2}$ ;
2.  $G = 2\mathbb{Z} = \{2z: z \in \mathbb{Z}\} = \{\dots, -4, -2, 0, 2, 4, \dots\}, a * b = a \cdot b$ ;
3.  $G = 2\mathbb{Z} + 1 = \{2z + 1: z \in \mathbb{Z}\} = \{\dots, -3, -1, 1, 3, \dots\}, a * b = a \cdot b$ ;
4.  $G = \{a + b\sqrt{2}: a, b \in \mathbb{Q}\}, a * b = a \cdot b$ ;
5.  $G = \mathbb{Q}, a * b = a + b + 5$ ;
6.  $G = \{a + b\sqrt{2}: a, b \in \mathbb{Q} \wedge (a, b) \neq (0, 0)\}, a * b = a \cdot b$ .

W części informatycznej napisz program, który będzie sprawdzał, czy podany zbiór skończony wraz z określonym w nim działaniem jest grupą (grupą abelową).

Zbiór i dwuargumentowe działanie w nim określone podane będzie za pomocą tablicy dwuwymiarowej, której postać jest analogiczna do postaci tabliczki mnożenia – element stojący w wierszu o numerze  $i$  oraz w kolumnie o numerze  $j$  jest wynikiem działania  $g_i * g_j$ , gdzie  $g_i$  i  $g_j$  są odpowiednimi elementami zbioru  $G$ .

Program po wczytaniu takiej tablicy (plik g.txt, w którym kolejne elementy w wierszach są oddzielone spacją, a kolejne wiersze są w nowych liniach) weryfikował i wypisywał będzie odpowiedni komunikat.

Przykładowo, dla tablic (oznaczenia  $*$ ,  $a, b, c, \dots$  dla wierszy i kolumn są jedynie w celach lepszej wizualizacji problemu, w pliku g.txt są jedynie wyniki):

*	a	b	c
a	a	b	c
b	c	b	a
c	a	b	c

*	a	b	c
a	a	a	a
b	a	a	a
c	a	a	a

*	a	b	c	d
a	a	b	c	d
b	b	a	d	c
c	c	d	a	b
d	d	c	b	a

*	a	b	c	d	f	g
a	a	b	c	d	f	g
b	b	a	d	c	g	f
c	c	f	a	g	b	d
d	d	g	b	f	a	c
f	f	c	g	a	d	b
g	g	d	f	b	c	a

program zwróciłby odpowiedzi: *nie jest grupą – brak łączności* (dla tabelki pierwszej); *nie jest grupą – brak elementu neutralnego* (dla tabelki drugiej); *jest grupą przemianą* (dla tabelki trzeciej); *jest grupą, ale nie jest grupą przemianą* (dla tabelki czwartej). Weryfikacja negatywna może być z różnych przyczyn, dla nas wystarczy, że program poda (poprawnie) jedną z tych przyczyn.



Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



Studenckie Koło Naukowo-  
Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



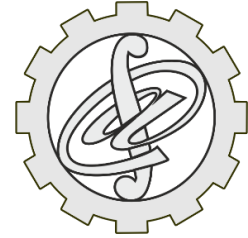
# POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej

Studenckie Koło Naukowo-Informatyczne „Link”

ul. Kaszubska 23, 44-100 Gliwice



## ZADANIE 4 – WĘDRÓWKI PO TABLICY

Niech  $n$  będzie dowolną liczbą naturalną większą od 3. Utwórzmy dwuwymiarową tablicę wymiaru  $n \times n$ , o elementach będących losowymi liczbami ze zbioru  $\{-1,0,1\}$ . Dla każdej z liczb tego zbioru poszukujemy ścieżki poziomej (łączącej pierwszą i ostatnią kolumnę) i ścieżki pionowej (łączącej pierwszy i ostatni wiersz). Przez ścieżkę będziemy rozumieć ciąg kolejnych elementów wygenerowanej tablicy, takich że w tablicy są one „sąsiadami” (również przez wierzchołek). Za każdą istniejącą ścieżkę dla danego elementu ze zbioru  $\{-1,0,1\}$  wygenerowana tablica otrzymuje jeden punkt, z tym że za ścieżkę poziomą (pionową) tablica otrzymuje tylko jeden punkt, nawet jeśli takich ścieżek dla danego elementu jest więcej niż jeden. Czyli maksymalnie tablica mogłaby (o ile jest to możliwe) otrzymać 6 punktów – za istnienie ścieżki poziomej dla  $-1$ ,  $0$  i  $1$  oraz za istnienie ścieżki pionowej dla każdego z tych elementów.

Napisz program, który dla podanej liczby naturalnej  $n$  generował będzie opisaną powyżej tablicę i wypisywał (formę wypisywania tablicy pozostawiamy w gestii rozwiązującego) będzie tę tablicę oraz odpowiadającą jej (określoną powyżej) liczbę punktów.

Przykładowo dla tablic:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & -1 \\ -1 & -1 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 & 1 \\ -1 & -1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 \\ -1 & -1 & -1 & 0 \\ -1 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & -1 & 1 \\ 0 & -1 & 0 & -1 & 1 & -1 \\ -1 & 1 & 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

otrzymalibyśmy odpowiedzi:

- dla tablicy pierwszej – 0 punktów;
- dla tablicy drugiej – 2 punkty – pozioma (kolor niebieski) ścieżka dla zera (poziomych ścieżek dla zera jest więcej, ale zgodnie z umową nie zwiększa się w takim przypadku liczba punktów) i pionowa ścieżka dla zera (niepogrubione zera);
- dla tablicy trzeciej – 2 punkty – pozioma (kolor pomarańczowy) ścieżka dla minus jedynek i pionowa ścieżka dla zera (kolor niebieski);
- dla tablicy czwartej – 4 punkty – pozioma (kolor pomarańczowy) ścieżka dla minus jedynek, pionowa ścieżka dla zera (kolor niebieski), pionowa ścieżka dla jedynek (kolor czerwony) i pionowa ścieżka dla minus jedynek (niepogrubione minus jedynek).

# ALGORYTMION

Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



Studenckie Koło Naukowo-Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



## POLITECHNIKA ŚLĄSKA

Dofinansowano ze środków Ministerstwa Edukacji i Nauki na podstawie umowy nr SONP/SP/550028/2022 z dnia 2022.10.28. Dofinansowanie w wysokości 116 836 zł, całkowita wartość projektu 135 580 zł.

Wydział Matematyki Stosowanej  
Studenckie Koło Naukowo-Informatyczne „Link”  
ul. Kaszubska 23, 44-100 Gliwice



### ZADANIE 5 – LITERALNIE

Ostatnio dość sporą popularnością cieszy pewna gra słowna, polegająca na dogadywaniu polskich (choć jest również wersja angielska – Wordle – zobacz stronę <https://www.nytimes.com/>) pięcioliterowych słów, w którą zagrać można na stronie <https://literalnie.fun>.

W tej grze podajemy istniejące w bazie słów pięcioliterowe słowo i dowiadujemy się, czy jest to prawidłowe słowo. Jeśli tak, to wygrywamy, a jeśli nie, to otrzymujemy informację o tym, czy pewne litery podanego przez nas słowa są na swoim miejscu (zaznaczone są one kolorem zielonym) czy też występują w poszukiwanym słowie, ale nie są na swoim miejscu (zaznaczone są one kolorem żółtym). Wszystkich prób mamy sześć, za każdym razem musimy podawać istniejące w bazie słowo oraz po każdej próbie otrzymujemy informację o wygranej lub o poprawności użytych liter. Gra posiada dwie wersje: w jednej z nich (wersja trudniejsza) za każdym razem (począwszy od ewentualnej drugiej próby) musimy podać istniejące słowo, w którym zaznaczone na zielono litery znajdują się na swoim miejscu, a litery zaznaczone na żółto występują w podawanym słowie, w wersji łatwiejszej wystarczy, że podawane słowo znajduje się w bazie (czy jest to wersja łatwiejsza, czy trudniejsza jest kwestią dyskusyjną).

Naszym zadaniem będzie napisanie programu, z którym użytkownik może rozegrać partię takiej gry. Dokładniej, program ze swojej bazy (bazę należy utworzyć na podstawie pliku *słownik.txt*, w którym słowa, każde w nowej linii, nie tylko pięcioliterowe, posortowane są alfabetycznie) wylosuje jedno słowo, zapyta gracza, którą wersję gry wybiera (trudniejszą, czy łatwiejszą) i poprowadzi do końca rozgrywkę.

Zakładamy, że w rozgrywce łatwiejszej gracz podaje za każdym razem słowo znajdujące się w utworzonej bazie słów pięcioliterowych (jeżeli gracz poda słowo innej długości lub spoza bazy słów otrzyma stosowny komunikat ale nie traci przy tym próby). W wersji trudniejszej, nie tylko powyższe założenia muszą być spełnione, ale również udzielane przez program wskazówki muszą zostać wykorzystane, tzn.:

- litera znajdująca się na swoim miejscu musi być w nowym słowie na tym miejscu;
- litera znajdująca się w poszukiwanym słowie, ale nie będąca na swoim miejscu, musi wystąpić w nowym słowie, ale nie może wystąpić na miejscu, o którym wiemy, że nie jest jej miejscem;
- litera, której nie ma w poszukiwanym słowie (w oryginalnej grze nie jest ona ani zielona, ani żółta) nie może wystąpić w nowym słowie.

Sposób komunikacji gracz–program, sposób podawania słów, formę informowania gracza o literach występujących na swoim miejscu i nie na swoim miejscu pozostawiamy w gestii rozwiązującego. W programie tym oceniana będzie również czytelność gry (przejrzystość rozgrywki).

## ALGORYTMION

Zespół „Algorytmion”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice



Studenckie Koło Naukowo-  
Informatyczne „Link”  
Politechnika Śląska  
Wydział Matematyki Stosowanej  
ul. Kaszubska 23  
44-100 Gliwice