ZADANIE 1 – LICZBY ZESPOLONE

Zadanie zaproponował: mgr Krzysztof Jarczewski, III LO im. S. Batorego w Chorzowie

Liczbą zespoloną nazywamy wyrażenie postaci $a + bi$, gdzie a i b są dowolnymi liczbami rzeczywistymi, a i jest jednostką urojoną, spełniającą warunek $i^2 = -1$ (jak widać, i nie może być liczbą rzeczywistą).

Napisz program, który po podaniu dwóch liczb zespolonych x i y oraz liczby naturalnej n, zwracał będzie sumę $x + y$, różnicę $x - y$, iloczyn $x \cdot y$, iloraz $\frac{x}{y}$ liczb zespolonych x i y oraz n-tą potęgę liczby zespolonej x.

Przykładowo, dla danych: $x = 3 + 2i$, $y = 2 - 3i$, $n = 3$, program zwróci kolejno: $5 - i$, $1 + 5i$, $12 - 5i$, i oraz $-9 + 46i$.

Wskazówka (do dzielenia liczb zespolonych): sprawdź, jaką ciekawą cechę ma iloczyn dwóch liczb zespolonych $a + bi$ oraz $a - bi$.
Sposób wprowadzania liczb zespolonych pozostawiamy w gestii rozwiązującego.

ZADANIE 2 – WALCEM PO WALCU

Zadanie zaproponowali: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska

Jeśli krzywą w przestrzeni zadamy równaniami:

$$\begin{cases} x = \cos t, \\ y = \sin t, \\ z = \varphi(t), \end{cases} \quad 0 \leq t \leq 2k\pi, \;\; k \in \mathbb{N},$$

to otrzymamy pewną krzywą, która będzie leżała na powierzchni bocznej walca o promieniu długości 1, którego oś OZ jest osią symetrii. Na przykład, jeśli $\varphi(t) = t$, to otrzymamy linię śrubową (gwint).

Załóżmy, że walec ten widzimy od strony punktu tej krzywej uzyskanego dla $t = 0$. Walec ten rozcinamy wzdłuż prostej prostopadłej do jego podstaw, biegnącej po jego powierzchni bocznej i przechodzącej przez punkt przeciwległy do tego, który ustalał widoczność. Następnie walec ten rozkładamy na płaszczyznę (prosta przeciwległa do prostej tnącej staje się osią OY), otrzymując w ten sposób prostokąt. Linia przestrzenna utworzy wewnątrz tego prostokąta pewną krzywą płaską.

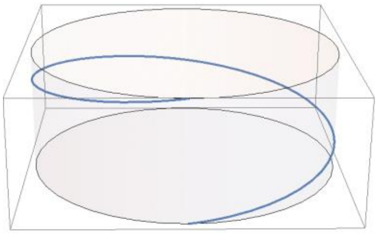
Napisz program, który dla zadanych argumentów: k i $\varphi(t)$ (sposób podawania i rodzaje funkcji φ pozostawiamy w gestii rozwiązującego) rysował będzie tę krzywą płaską.

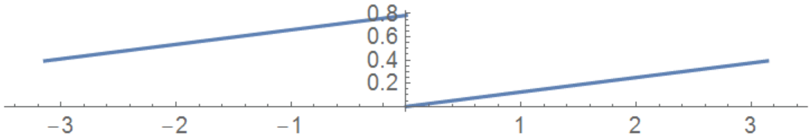
Przykłady (rozwiązaniami są rysunki po prawej stronie):

POLITECHNIKA ŚLĄSKA

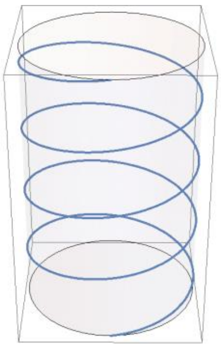
Wydział Matematyki Stosowanej
Studenckie Koło Naukowo-Informatyczne „Link”
ul. Kaszubska 23, 44-100 Gliwice

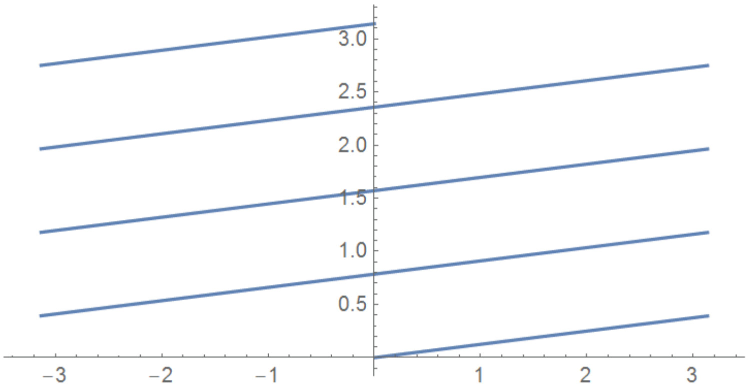
$k = 1, \varphi(t) = t/8$:



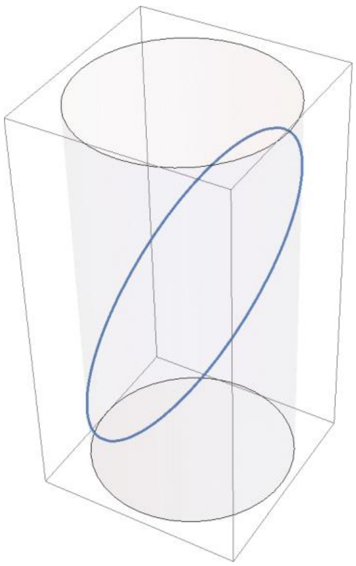


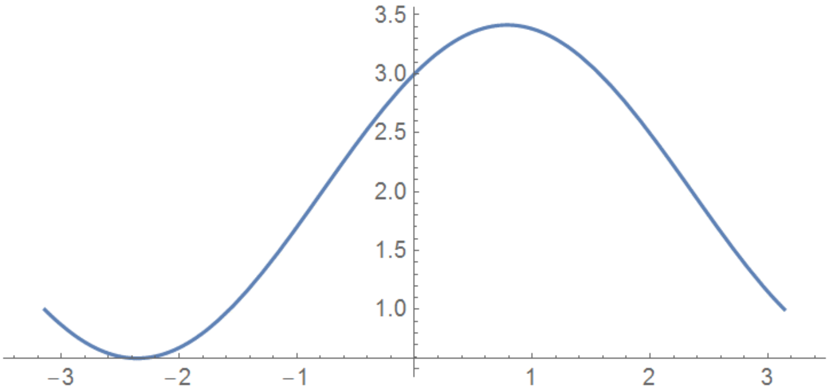
$k = 4, \varphi(t) = t/8$:





$k = 1, \varphi(t) = 2 + \sin t + \cos t$:





ZADANIE 3 – TATARAKI I BALONIK

Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska (zadanie finałowe edycji 2016/17)

W pliku *słownik.txt* znajduje się słownik, w którym słowa (każde w nowej linii) posortowane są alfabetycznie.

Napisz program, który dla zadanej początkowej litery słowa, poszukiwał będzie w tym słowniku wyrazów, które da się podzielić na pewnym miejscu w ten sposób, że zarówno do miejsca podziału jak i od miejsca podziału (tę część czytamy wspak), tak powstałe słowa również znajdują się w tym słowniku.

Przykładowo, jeśli podalibyśmy jako argument literę *t*, to program mógłby znaleźć słowo *tataraki*, bo dzieląc je po czwartej literze, otrzymamy słowa *tata* i *ikar*, a dla litery *b*, program mógłby zwrócić słowo *balonik* (podział po trzeciej literze na słowa *bal* i *kino*).

Zakładamy dodatkowo, że zarówno poszukiwane słowo, jak i jego składowe, są co najmniej dwuliterowe.

Program ma zwracać wszystkie wyrazy spełniające warunki zadania (na zadaną literę początkową).

ZADANIE 4 – ROZKŁAD

Zadanie zaproponowali: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska

Dana jest dwuwymiarowa tablica T_{ij}, $t_{ij} \in \mathbb{R}, 1 \le i,j \le n, n \in \mathbb{N}, n > 1$. Zdefiniujmy dla takich tablic pewne działanie $*$: $T_{ij} * R_{ij} = P_{ij}$, gdzie:

$$p_{ij} = \sum_{k=1}^{n} t_{ik} \cdot r_{kj}\,,\ \ 1 \le i,j \le n.$$

Na przykład, jeśli:

$$T = \begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & -3 \\ -1 & 0 \end{bmatrix},$$

to

$$T * R = P = \begin{bmatrix} -3 & 3 \\ -1 & 0 \end{bmatrix},$$

bo, przykładowo, $p_{1\,2} = t_{1\,1} \cdot r_{1\,2} + \ t_{1\,2} \cdot r_{2\,2} = -1 \cdot (-3) + 2 \cdot 0 = 3 + 0 = 3.$

Okazuje się, że często taką dwuwymiarową tablicę T_{ij} da się jednoznacznie przedstawić w postaci $T_{ij} = D_{ij} * G_{ij}$, gdzie tablice D_{ij} i G_{ij} mają szczególną postać, a mianowicie $d_{ij} = 0$ dla $i < j$ oraz $g_{ij} = 0$, jeśli $i > j$ i $g_{ij} = 1$ dla $i = j$, na przykład:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & -1 \\ 4 & 3 & -4 & -11 \\ 7 & 6 & -4 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & -3 & 0 & 0 \\ 4 & -5 & -6 & 0 \\ 7 & -8 & -9 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Napisz program, który dla zadanej dwuwymiarowej tablicy (sposób wprowadzania tablicy pozostawiamy w gestii rozwiązującego) zwróci na ekran, najlepiej w postaci dwóch dwuwymiarowych tablic, odpowiednie tablice D_{ij} i G_{ij}.

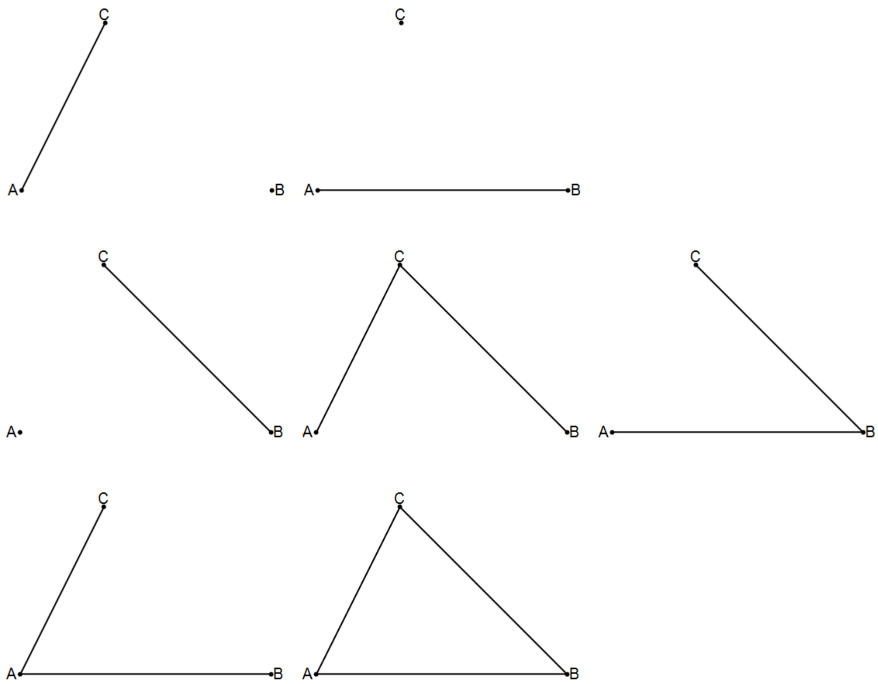
ZADANIE 5 – MIASTA I DROGI

Zadanie zaproponowali: dr inż. Adam Zielonka i dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska

Danych jest $m > 2$ różnych miast, o których zakładamy, że żadne trzy z nich nie leżą na jednej prostej. Między każdymi dwoma miastami można wybudować drogę będącą odcinkiem (i tylko taką).

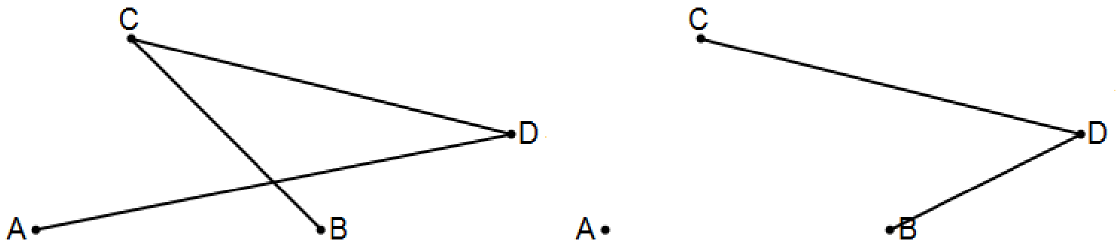
Napisz program, który dla zadanej wartości $2 < m \in \mathbb{N}$, zwracał będzie liczbę różnych sposobów wybudowania dróg między tymi miastami, tak aby dla każdej trójki miast, spośród wszystkich m miast, istniały co najmniej dwa miasta połączone drogą. Program ma zwracać również najmniejszą liczbę takich dróg.

Na poniższym rysunku przedstawione są wszystkie rozwiązania dla $m = 3$.



Jak widać, jest tu 7 możliwości, a najmniejsza liczba dróg wynosi 1.

Poniżej przykład poprawnego rozwiązania (po lewej) i niepoprawnego (po prawej) dla $m = 4$.



W tym przypadku najmniejsza liczba dróg wynosi 2 (nie jest ona przedstawiona na tym rysunku).