

ZADANIE 1 – WYKREŚŁ „ANKA”

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

Jednym z prostszych zadań szaradziarskich jest wykreślanka. Mamy tam podaną prostokątną planszę wypełnioną literami i listę słów. Rozwiązanie zadania polega na znalezieniu na planszy wszystkich podanych wyrazów (ich kolejne litery muszą ze sobą sąsiadować). Po wykreśleniu znalezionych wyrazów (trzeba znaleźć wszystkie) pozostanie na planszy pewna liczba niewykreślonych liter, które utworzą hasło – rozwiązanie zadania.

W pliku *plansza.txt* podany jest zestaw liter, które tworzą planszę – litery w wierszach nie są oddzielane spacjami, ale każdy wiersz jest w nowej linii. W pliku *wyrazy.txt* znajdują się poszukiwane (na planszy) wyrazy – każdy wyraz w nowym wierszu.

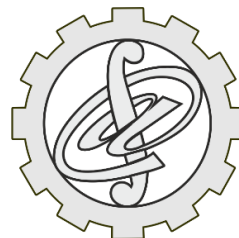
Zakładamy, że wyrazy na planszy mogą się znajdować w wierszu, w kolumnie, lub „po skosie” i ułożone mogą tam być od lewej do prawej albo od prawej do lewej (w wierszu) lub od góry do dołu albo od dołu do góry (w kolumnie), a po skosie również mogą być w każdym ułożeniu. Położenie każdego z wyrazów w planszy jest jednoznaczne. Każda litera z planszy może być użyta tylko raz (wchodzi tylko raz do jednego z wyrazów lub nie wchodzi w ogóle).

Napisz program, który po wczytaniu plików *plansza.txt* i *wyrazy.txt* rozwiązywał będzie tak zadaną wykreślankę, tzn. poszukiwał będzie wszystkich podanych wyrazów, a na koniec, czytając niewykorzystane litery (wierszami od góry go dołu, a wewnątrz wiersza od lewej do prawej) wypisze hasło.

Przykładowo, jeśli plansza miałaby postać przedstawioną poniżej (kolory i linie stanowią jedynie wizualizację) i podane byłyby wymienione poniżej wyrazy:

T	G	O	M	S	K	C	A	bilet	grzywka	T	G	O	M	S	K	C	A
E	T	U	L	O	D	Ś	K	jura	kometa	E	T	U	L	O	D	Ś	K
L	O	E	M	E	G	N	W	leń	raróg	L	O	E	M	E	G	N	W
I	A	E	F	Ó	Ń	I	Y	śnieg	teflon	I	A	E	F	Ó	Ń	I	Y
B	T	R	R	L	Z	E	Z	zęb		B	T	R	R	L	Z	E	Z
A	W	A	U	Ą	O	G	R			A	W	A	U	Ą	O	G	R
N	R	I	B	J	E	N	G			N	R	I	B	J	E	N	G

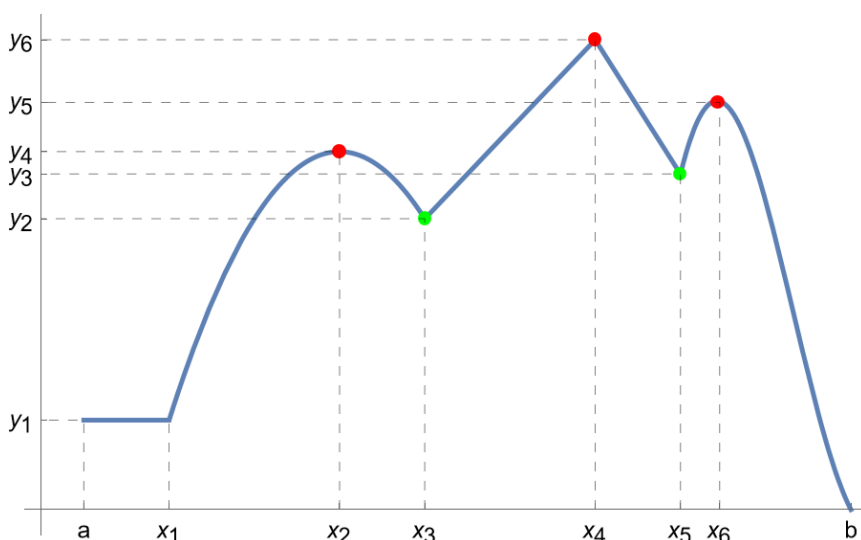
program zwróciłby odpowiedź: *cudownie* (w zwracanej odpowiedzi oraz w obydwu plikach: z planszą i z wyrazami nie zwracamy uwagi na wielkość liter).



ZADANIE 2 – EKSTRA EKTREMUM

Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska (zadanie z finału konkursu Algorytmion, edycja 2023/2024)

Maksimum (minimum) lokalnym funkcji f nazywamy taki punkt x_0 , dla którego istnieje takie $\delta > 0$, że $(x_0 - \delta, x_0 + \delta) \subset D_f$ oraz że w przedziale $(x_0 - \delta, x_0 + \delta)$ funkcja f ma wartość największą (najmniejszą) w punkcie x_0 .



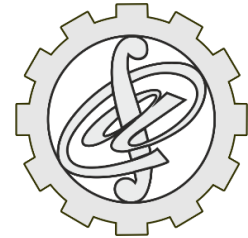
Na rysunku obok naszkicowany jest wykres pewnej funkcji f dla $x \in [a, b]$, dla którego kolorem czerwonym zaznaczone są punkty, będące maksimum lokalnym funkcji f , zaś kolorem zielonym – punkty, będące minimami lokalnymi tej funkcji. Można więc wyobrazić sobie maksima jako lokalne „górki” na wykresie danej

funkcji, a minima jako lokalne „doliny”. Często w definicji maksimów/minimów danej funkcji dopuszczalne są również punkty, w który wartość nie koniecznie jest największa (najmniejsza), ale wystarczy, że nie ma od niej wartości większej (mniejszej), w takim przypadku przedstawiona powyżej funkcja miałaby nieskończenie wiele maksimów i minimów lokalnych (np. punkt (x_1, y_1) byłby wtedy minimum lokalnym). My jednak pozostaniemy przy pierwotnej definicji. Dodatkowo warto też zauważyć, że również żaden z punktów $(a, y_1), (b, 0)$ nie jest ani maksimum, ani minimum tej funkcji.

Niech f będzie funkcją postaci:

$$f(x) = \sum_{i=1}^n T_i(V_i^m(x)),$$

gdzie T_i jest jedną z dwóch funkcji: sinus lub kosinus, $n \geq 1$ jest daną liczbą naturalną, a V_i^m jest pewnym wielomianem stopnia $m \geq 1$.



Przykładowo, dla $n = m = 2$ przyjmując T_1 jako sinus, a T_2 jako kosinus, funkcja f mogłaby mieć postać:

$$f(x) = \sin(2x^2 - 3x + 1) + \cos(-x^2 + x + 2).$$

Napisz program, który dla danych: $n, m, \delta, x_1, x_2, \dots, x_{2k-1}, x_{2k}$, przy czym zmienne te oznaczają odpowiednio: liczbę składników funkcji f , stopień wszystkich wielomianów, dokładność, końce przedziałów składających się na dziedzinę funkcji f , a dokładniej $D_f = [x_1, x_2] \cup [x_3, x_4] \cup \dots \cup [x_{2k-1}, x_{2k}]$, poszukiwał będzie maksimum i minimum funkcji f (z dokładnością δ).

Poszukiwania te opierać się będą na losowości – program wielokrotnie losował będzie z dziedziny funkcji f punkt, który weryfikowany będzie, czy spełnia warunek bycia maksimum (minimum). Oczywiście taka weryfikacja nie będzie zgodna z definicją, wystarczy np. sprawdzić, czy dla podanej wartości δ spełnione są warunki: $f(x_0) > f(x_0 - \delta)$ i $f(x_0) > f(x_0 + \delta)$ (dla maksimum). Podobnie losowo wybierane będą współczynniki generowanych wielomianów V_i^m oraz rodzaje funkcji T_i , gdzie $i = 1, 2, \dots, m$. Zakres i typ liczb będących współczynnikami wielomianów V_i^m pozostawiamy w gestii rozwiązującego (w przykładowej, podanej powyżej funkcji f były to liczby całkowite z zakresu od -3 do 3).

Finalnie program wypisywał będzie (na ekran lub do pliku) odpowiedź postaci (na przykładzie podanej powyżej funkcji f):

maksima lokalne:

w punkcie -1.943 wartość 0.1325

w punkcie -1.212 wartość 1.738

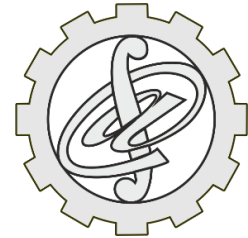
w punkcie -0.27 wartość 0.84

minima lokalne:

w punkcie -1.642 wartość -1.642

w punkcie -0.763 wartość -0.173

w punkcie 0.678 wartość -0.717



ZADANIE 3 – METAMORFOZA

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

Jednym z zadań szaradziarskich jest *metamorfoza* (zobacz zadanie 5 z eliminacji 2023/2024), gdzie podane są dwa wyrazy o tej samej liczbie liter i takich, że żadna z liter słowa pierwszego nie jest literą drugiego słowa. Następnie należy wymieniać po jednej literze danego słowa (w pierwszym kroku jest to pierwsze słowo, w drugim – słowo powstałe przez wymianę w pierwszym słowie jednej litery, w trzecim – słowo powstałe przez wyminę niewymienionej do tej pory litery w słowie drugim, itd.) Przed wpisaniem kolejnego słowa litery można przestawiać miejscami (anagramować). Każda litera pierwszego słowa musi być zamieniona i każda litera drugiego z danych słów musi zostać wykorzystana.

Jednym z wariantów metamorfozy jest odmiana, w której wewnętrzne słowa są wydłużone o dodatkowe: początkową i końcową literę (litery te są niezależne od podanych skrajnych słów).

Zadanie mogłoby przyjąć następującą formę (po lewej treść zadania, po prawej rozwiązanie z pomocniczym kolorowaniem liter):

	R	E	C	E	
P					Z
K					E
K					Ć
	N	O	G	I	

→

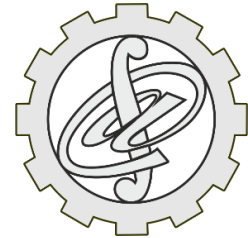
	R	E	C	E	
P	O	R	E	C	Z
K	R	O	C	I	E
K	N	O	C	I	Ć
	N	O	G	I	

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

Napisz program, który dla danego $2 < n \in \mathbb{N}$ poszukiwał będzie słów znajdujących się w słowniku, dla których istnieje „metamorfoza” jednego z tych słów w drugie. Oczywiście słowa „pośrednie” również muszą znajdować się w tym słowniku.

Program ma mieć trzy możliwości działania: w pierwszej działa jak w powyższym akapicie, w drugim podajemy mu obydwa skrajne wyrazy i kolejne pary skrajnych liter, a w trzecim jedynie pierwszy i ostatni wyraz.

Program po wprowadzeniu odpowiednich argumentów poszukiwał będzie (i wypisywał) odpowiednich słów rozwiązujących metamorfozę, a jeżeli nie znajdzie takich słów, to wypisze komunikat, że takich słów nie ma.



ZADANIE 4 – ZNIEWOLENIE? – NIE!

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

W pliku *ciag_znakow.txt* znajduje się ciąg znaków złożony jedynie z małych liter alfabetu łacińskiego.

Napisz program, który wypisze najdłuższy podciąg tego ciągu znaków, którego indeksy znaków występują bezpośrednio po sobie i który się powtarza w tym ciągu (powtarzający się ciąg również ma indeksy znaków występujące bezpośrednio po sobie). Program podaje również indeksy występowania tego podciągu.

Przykładowo, dla ciągu znaków: *sesjaegzaminacyjna* program zwróci podciąg *na* i zakresy indeksów 12-13 oraz 17-78.

Dla ciągu znaków: *rabarbar* program zwróci podciąg *bar* i zakresy wystąpienia jego indeksów 3-5 oraz 6-8.

Dla ciągu znaków: *abbbaswbbsdsdssdssbbaaw* program zwróci podciąg *bbb* i zakresy wystąpienia jego indeksów 2-4 oraz 8-10, ale też podciąg *dss* i zakresy wystąpienia jego indeksów 15-17 oraz 18-20.

ZADANIE 5 – CIĄG, W KTÓRYM WAŻNĄ ROLĘ GRA FUNKCJA θ

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

Dana jest liczba naturalna k . Oznaczmy przez $\theta(k)$ liczbę wszystkich naturalnych dzielników liczby k . Określmy ciąg:

$$a_n = \begin{cases} k, & n = 1, \\ \theta(a_{n-1}), & n > 1. \end{cases}$$

Udowodnij, że ciąg ten, dla dowolnej liczby naturalnej k , począwszy od pewnego miejsca, jest ciągiem stałym.

Napisz program, który dla zadanej liczby naturalnej k wypisywał będzie wszystkie kolejne wyrazy ciągu a_n aż do wyrazu, od którego ciąg ten staje się ciągiem stałym.

Przykładowo, jeśli podamy $k = 24$, to program zwróciłby ciąg:

24, 8, 4, 3, 2.