

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE

Wykład dla kierunku:
Matematyka stosowana
i technologie
informatyczne

PROWADZI:

Jarosław Bilski
Katedra Inteligentnych Systemów Informatycznych
d. Katedra Inżynierii Komputerowej
Politechnika Częstochowska

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 2

ZAJĘCIA:

- Wykład 2 godziny tygodniowo
- Laboratorium 2 godziny tygodniowo

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 3

PODSTAWOWA LITERATURA:



- Zbigniew Czech: Wprowadzenie do obliczeń równoległych, PWN 2010,
- Maurice Herlihy, Nir Shavit: Sztuka programowania wieloprocesorowego, PWN 2010,
- Jason Sanders, Edward Kandrot: CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU, Helion 2012
- Eugeniusz Wróbel – Praktyczny kurs asemblera, wyd. II, Helion 2011
- Intel® 64 and IA-32 Architectures Software Developer's Manual

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 4

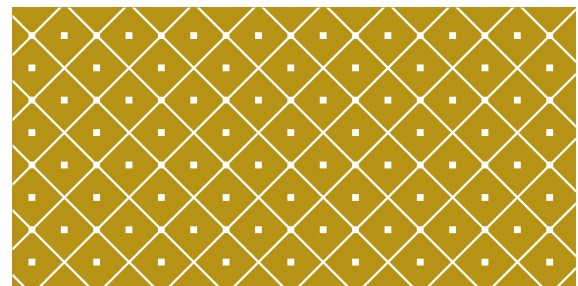
LITERATURA DODATKOWA:



- Marek Sawewain: OpenCL. Akceleracja GPU w praktyce, PWN 2014,
- Wład Pirogow: Asembler. Podręcznik programisty, Helion 2005
- Roman Wyrzykowski: Klastry komputerów PC i architektury wielordzeniowe: budowa i wykorzystanie, AOWE:xit 2009

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 5



WPROWADZENIE DO PRZETWARZANIA WEKTOROWEGO I RÓWNOLEGŁEGO

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 6

PO CO TO WSZYSTKO?

- Rozwiązujemy coraz bardziej złożone problemy.
- Realizujemy obliczenia na wektorach i macierzach.
- Mamy coraz więcej danych.
- Chcemy uzyskiwać wyniki w krótszym czasie.
- Taktowanie procesorów rośnie powoli.
- Mamy sprzęt pozwalający na obliczenia równoległe.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 7

TAKSONOMIA FLYNNA

Flynn zaklasyfikował programy i komputery w zależności od tego, czy dany program lub komputer korzysta z jednego, czy z wielu zbiorów instrukcji oraz czy te instrukcje korzystają z jednego, czy z wielu zbiorów danych.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 8

TAKSONOMIA FLYNNA

Flynn wyróżnił cztery klasy:

SISD (ang. single-instruction-single-data) równoważną przetwarzaniu całkowicie sekwencyjnemu;

SIMD (ang. single-instruction-multiple-data), gdzie wykonuje się te same operacje na różnych zbiorach danych;

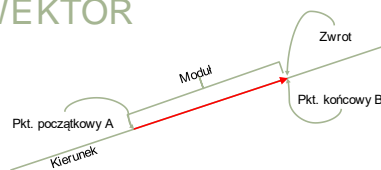
MISD (ang. multiple-instruction-single-data) wykonuje się różne operacje na tym samym zbiorze danych (systolic arrays);

MIMD (ang. Multiple-instruction-multiple-data), gdzie różne operacje wykonywane są na różnych zbiorach danych.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 9

WEKTOR



Wektor - to uporządkowana para punktów – punkt początkowy (zaczepienia) i końcowy.

Wektor cechuje: moduł (długość, wartość), kierunek (kierunek prostej zawierającej wektor) i zwrot.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 10

WEKTOR

Wektor może być **zaczepiony** lub **swobodny**.

W wektorze swobodnym punkt zaczepienia jest nieistotny.

Współrzędne końca wektora umieszczonego w początku układu współrzędnych tworzą składowe lub współrzędne wektora.

Wektory oznaczane są jako: \overrightarrow{AB} , \mathbf{a} , \vec{a} , $\mathbf{a} = \begin{bmatrix} 3 \\ 1 \\ 8 \\ 5 \end{bmatrix}$.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 11

PROGRAMOWANIE WEKTOROWE

Programowanie wektorowe polega na wykonywaniu operacji na całych wektorach (lub ich fragmentach) zamiast na skalarach. Wymagany do tego jest sprzęt umożliwiający wykonywanie operacji na wielu danych jednocześnie (procesor wektorowy).

Możliwości takie posiadają współczesne procesory CPU z instrukcjami SIMD oraz procesory graficzne GPU.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 12

PROGRAMOWANIE WEKTOROWE

Przykładowo dodawanie wektorów:

$$\begin{bmatrix} 4 \\ 5 \\ 9 \\ 8 \end{bmatrix} := \begin{bmatrix} 3 \\ 1 \\ 8 \\ 5 \end{bmatrix} + \begin{bmatrix} 1 \\ 4 \\ 1 \\ 3 \end{bmatrix}$$

odbywa się przy użyciu jednej instrukcji zamiast czterech. W przypadku wektorów o dużej liczbie wymiarów może wystąpić konieczność podzielenia wektorów na fragmenty w zależności od możliwości sprzętu.

PROGRAMOWANIE WSPÓLBIEŻNE

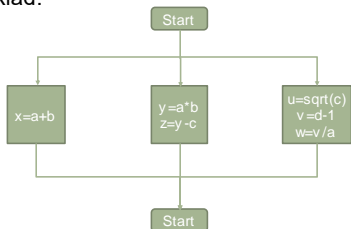
Programowanie współbieżne polega na podziale programu na wątki, które będą wykonywane na różnych procesorach fizycznych lub logicznych.

Procesory te mogą należeć do jednego systemu komputerowego, do klastra lub grid'a.

Wątki mogą wykonywać różne zadania i współdzielić procesor z innymi wątkami.

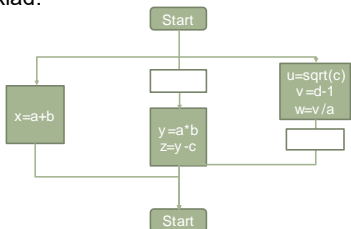
PROGRAMOWANIE WSPÓLBIEŻNE

Przykład:



PROGRAMOWANIE WSPÓLBIEŻNE

Przykład:



PROGRAMOWANIE RÓWNOLEGŁE

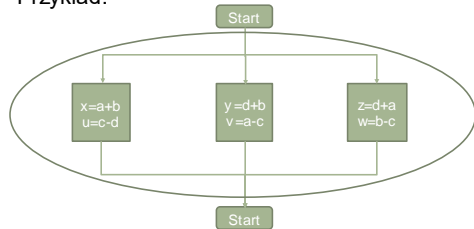
Programowanie równoległe polega na takim przygotowaniu programu i danych, aby operacje były wykonywane synchronicznie dokładnie w tym samym czasie.

Procesory muszą należeć do jednego systemu komputerowego i być synchronizowane.

Wątki nie mogą współdzielić procesora z innymi wątkami.

PROGRAMOWANIE RÓWNOLEGŁE

Przykład:



PROGRAMOWANIE MIESZANE WSPÓLBIEŻNO- RÓWNOLEGŁE

Programowanie mieszane występuje dla dużych rozmiarów/zbiorów danych. Procesory wykonują swoje obliczenia równoległe i współbieżnie z innymi procesorami.

Procesory nie muszą należeć do jednego systemu komputerowego.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 19

ZASOBY NIEPODZIELNE, SEKCJA KRYTYCZNA, WZAJEMNE WYKLUCZANIE, SYNCHRONIZACJA I ZAKLESZCZENIE

Współbieżne wątki mogą odwoływać się do zmiennej niepodzielnej, co może powodować, że wynik działania programu będzie nieprawidłowy. Zjawisko to jest nazywane **wyścigami** lub **hazardem**, a fragment kodu, odwołujący się do tej zmiennej, nazywa się **sekcją krytyczną**. Należy zapewnić **wzajemne wykluczenie** pomiędzy wątkami w dostępie do takiej zmiennej.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 20

ZASOBY NIEPODZIELNE, SEKCJA KRYTYCZNA, WZAJEMNE WYKLUCZANIE, SYNCHRONIZACJA I ZAKLESZCZENIE

Wzajemne wykluczenie można zrealizować używając np. semaforów.

Pozwalają one na synchronizację wątków, jednak nieprawidłowo zastosowane mogą prowadzić do zakleszczenia wątków i tym samym nie pozwalają zakończyć obliczeń.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 21

PROBLEM AKUMULATORA

Iloczyn skalarny:

$$\begin{bmatrix} 3 \\ 1 \\ 8 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 4 \\ 1 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 4 \\ 8 \\ 15 \end{bmatrix} \rightarrow 3 + 4 + 8 + 15 \rightarrow 30$$

Wszystkie mnożenia mogą odbyć się równoległe w tym samym czasie. 1op.

Dodawanie odbywa się szeregowo. 3 op. Lub ogólniej n-1 operacji.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 22

ZASTOSOWANIE REDUKCJI

Proces redukcji polega na zastąpieniu ciągu operacji realizowanych sekwencyjnie nowym ciągiem operacji, w którym w kolejnych krokach wykonuje się maksymalnie dużo operacji jednocześnie.

Redukcja prowadzi do zmniejszenia liczby kroków, liczba operacji się nie zmniejsza.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 23

ZASTOSOWANIE REDUKCJI

Iloczyn skalarny z zastosowaniem redukcji:

$$\begin{bmatrix} 3 \\ 1 \\ 8 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 4 \\ 1 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 4 \\ 8 \\ 15 \end{bmatrix} \rightarrow \begin{bmatrix} 7 \\ 23 \end{bmatrix} \rightarrow 30$$

Wszystkie mnożenia mogą odbyć się równoległe w tym samym czasie. 1op.

Dodawanie odbywa się parami. 2 op. Lub ogólniej $\text{roundup}(\log_2 n)$ operacji.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 24

WARUNKI BERNSTEINA

Dla dwóch fragmentów programu P_i i P_j . Warunki Bernsteina mówią, kiedy te dwa fragmenty są niezależne od siebie i mogą być wykonane równoległe bez żadnych przeszkód.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 25

WARUNKI BERNSTEINA

Dwa działania (podprogramy, operacje) i oraz j są niezależne i mogą zostać wykonane równoległe gdy:

$$I_i \cap O_j = \emptyset$$

$$I_j \cap O_i = \emptyset$$

$$O_i \cap O_j = \emptyset$$

gdzie:

I_i, I_j - zbiór zmiennych wejściowych i/j -tego działania

O_i, O_j - zbiór zmiennych wyjściowych i/j -tego działania

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 26

WARUNKI BERNSTEINA

Niespełnienie pierwszego lub drugiego warunku powoduje zależność przepływu, co oznacza, że wynikiem działania P_i jest wartość wykorzystywana przez P_j (lub odwrotnie).

Trzeci warunek wymaga niezależności wyjść. Niespełnienie tego warunku powoduje, że dwie wartości są zapisywane w jednym miejscu pamięci i druga nadpisze pierwszą.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 27

SPOWOLNIENIE RÓWNOLEGŁE

Jeśli program jest dzielony na coraz więcej wątków, to może się okazać, że narzuty związane z komunikacją i dostępem do pamięci zaczną przekraczać zysk wynikający ze „zrównoleglenia” i pomimo zwiększania potencjalnej mocy obliczeniowej uzyskuje się spowolnienie obliczeń.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 28

PRAWO AMDAHLA

jest używane do określania maksymalnego spodziewanego zwiększenia wydajności całkowitej systemu jeżeli tylko część systemu została ulepszona/zrównoleglona. Jest ono używane w obliczeniach równoległych do przewidzenia teoretycznego maksymalnego wzrostu szybkości obliczeń przy użyciu wielu procesorów.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 29

PRAWO AMDAHLA

Dla obliczeń współbieżnych:

$$S_C = \frac{1}{1 - P + \frac{P}{S}}$$

- S_C – maksymalne przyspieszenie całkowite algorytmu
- P – część możliwa do zrównoleglenia algorytmu
- S – maksymalne przyspieszenie części równoległej

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGŁE 30

PRAWO AMDAHLA

Dla obliczeń równoległych na N procesorach:

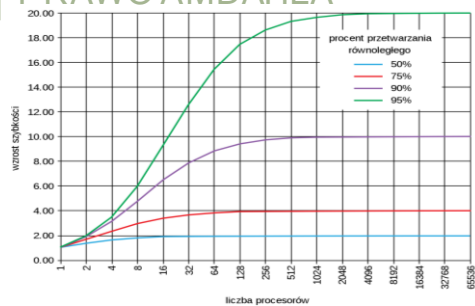
$$S_C = \frac{1}{1 - P + \frac{P}{N}}$$

- S_C – maksymalne przyśpieszenie całkowite algorytmu
- P – część możliwa do zrównoleglenia algorytmu
- N – liczba procesorów

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 31

PRAWO AMDAHLA



(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 32

PRAWO AMDAHLA

Graniczne przyśpieszenie wynosi:

$$S_C = \frac{1}{1 - P}$$

- S_C – maksymalne przyśpieszenie całkowite algorytmu
- P – część możliwa do zrównoleglenia algorytmu

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 33

PRAWO GUSTAFSONA

Czas obliczeń:
równoległych $T_p = \alpha + \beta = 1$

Szeregowych $T_s = \alpha + N\beta$

Przyśpieszenie $S = \alpha + N\beta = \alpha + N(1 - \alpha)$

- S – przyśpieszenie
- N – ilość procesorów
- α, β – udział części szeregowej/równoległej w rozwiązaniu równoległym

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 34

PRAWO GUSTAFSONA

$$S(N) = N - \alpha \cdot (N - 1)$$

- S – przyśpieszenie
- N – ilość procesorów
- α – udział części szeregowej w rozwiązaniu równoległym

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 35

PRAWO GUSTAFSONA

Mówi, że zmniejszając udział części szeregowej lub zwiększając złożoność problemu może zwiększyć przyśpieszenie algorytmu równoległego.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 36

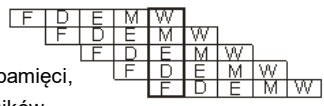
CYKL OBLICZENIOWY

Cykl obliczeniowy to jest czas potrzebny na wykonanie pojedynczej operacji obliczeniowej (dodawanie, mnożenie, obliczenie funkcji) lub wielu pojedynczych niezależnych operacji w sposób równoległy.

RÓWNOLEGŁE WYKONYWANIE INSTRUKCJI

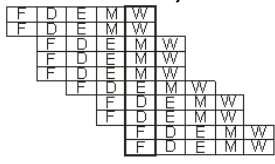
Wykonywanie instrukcji w **potoku**. Fazy:

- F - pobieranie instrukcji,
- D - dekodowanie,
- E - wykonanie,
- M - odwołanie do pamięci,
- W - zapisanie wyników.



RÓWNOLEGŁE WYKONYWANIE INSTRUKCJI

Superskalarność – cecha procesorów oznaczająca możliwość wykonywania kilku instrukcji (rozkazów maszynowych) jednocześnie, uzyskiwana poprzez zwielokrotnienie jednostek wykonawczych



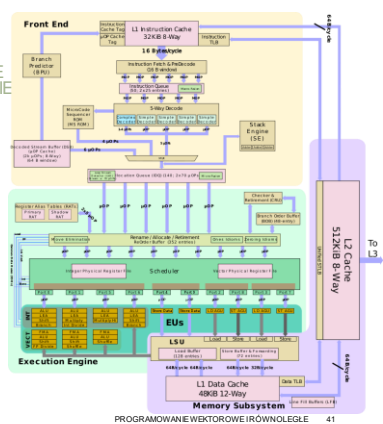
RÓWNOLEGŁE WYKONYWANIE INSTRUKCJI

Wykonywanie poza kolejnością (out-of-order execution) – zdolność mikroprocesora superskalarnego do zmiany kolejności wykonywania instrukcji, tak, aby maksymalnie wykorzystać jego jednostki wykonawcze (obliczeniowe) i równoległe, w kilku potokach, wykonać jak najwięcej instrukcji. Powoduje to przyspieszenie wykonywania programów.

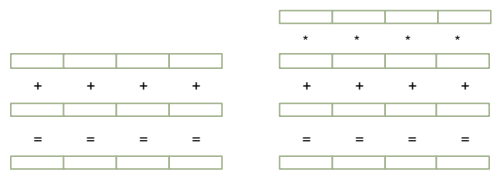
Zmiana kolejności jest możliwa tylko wówczas, gdy instrukcje są od siebie niezależne.

RÓWNOLEGŁE WYKONYWANIE INSTRUKCJI

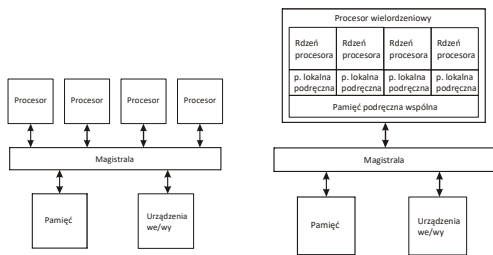
Porty w procesorze.



INSTRUKCJE SIMD: AVX I FMA



SYSTEMY WIELOPROCESOROWE SMP

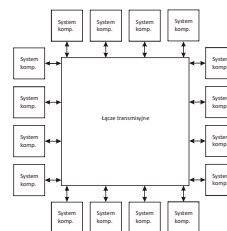


(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 43

SYSTEM WIELOKOMPUTEROWY

Klaster lub grid

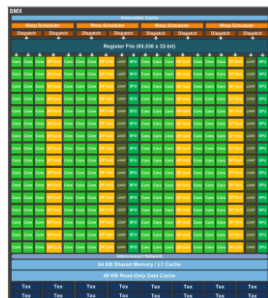


(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 44

GPGPU – KARTY GRAFICZNE

Architektura Kepler - NVIDIA



- Moduł SMX zawiera:
- 192 rdzenie pojedynczej precyzji
 - 64 rdzenie podwójnej precyzji
 - 32 jednostki specjalne,
 - 32 jednostki odczytująco-zapisujące.

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 45

PRZYKŁAD

iloczyn skalarny

Scenariusz	Zadanie	Skala pomiarowa	Tabela pomiarowa
1024	L.O	1000000	33
Wynik			
253.322791312038		2796	9132
253.322791312038		866	2742
Wskazniki wsm			
PPU[0-3]	253.322791312038	865	2736
PPU[4-7]	253.322791312038	431	1363
PPU[8-11]	253.322791312038	357	1128
GPU[0-2]	253.322791312038	430	1371
GPU[3-5]	253.322791312038	225	675
GPU[6-8]	253.322791312038	284	840
GPU[9-12]	253.322791312038	298	871
Funkcje w bibliotece DLL			
PPU[0-3]	253.322791312038	866	2742
PPU[4-7]	253.322791312038	433	1363
PPU[8-11]	253.322791312038	358	1124
GPU[0-2]	253.322791312038	434	1262
GPU[3-5]	253.322791312038	225	675
GPU[6-8]	253.322791312038	288	852
TaskId: 433			
AVX[0-3]	253.322791312038	249	742
AVX[4-7]	253.322791312038	167	495
AVX[8-11]	253.322791312038	157	507
AVX[12-15]	253.322791312038	433	1283
AVX[16-19]	253.322791312038	226	681
AVX[20-23]	253.322791312038	187	576

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 46

PRZYKŁAD

Rekonstrukcja obrazu – tomograf komputerowy

Opis programu	Ilość wątków	1 iteracja [ms]	20k iteracji [s]	Przyspieszenie
Oryginalny	1	688	13760 3h49m20s	-
Asembler x64	1	8,1	162 2m42s	84,938
Asembler x64 wielowątkowy	8	1,119047	22,38095	614,809
i9-7900X	10	0,994545	19,89089	691,774
10r 20t	16	1,017738	20,35476	676,009
	20	0,930187	18,60375	739,636
i9-9980X 18r	16	0,68025	13,605s	1011,392
NVIDIA 1080Ti - 3584r		1,74473	34,8946	394,330
NVIDIA Titan V – 5120r		0,70534	14,091075	975,416

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 47

NIESPODZIANKA

Losuj	Rozmiar	Zakres	Ilość powtórzeń	Czas pomiaru		
				Czas dodawania	Czas mnożenia	Czas peta/iloczyn skalarny
4	1000	1,0	1000000	490	490	250
1	710	1270	710			
4	240	300	300			

(C) KISI d.KIK PCz 2023

PROGRAMOWANIE WEKTOROWE I RÓWNOLEGLE 48