

Zadanie 1: symulacja rozchodzenia się ciepła

Naszym celem będzie napisanie funkcji symulujących fikcyjny przepływ pewnej wartości pseudofizycznej (o wartościach z dziedziny liczb rzeczywistych) przez prostokątną siatkę. Dla ułatwienia nazwijmy tę wartość ciepłem. Nasza symulacja będzie oczywiście bardzo uproszczona, ale przypominająca tzw. metody elementów skończonych.

Dana jest prostokątna siatka w postaci tablicy dwuwymiarowej (oczywiście program może na swoje potrzeby trzymać to w innej postaci, ale mówimy o zewnętrznej, „logicznej” reprezentacji). W każdej komórce przechowujemy bieżącą wartość temperatury w tej okolicy.

W niektórych miejscach siatki znajdują się grzejniki -- źródła ciepła o ustalonej na czas symulacji temperaturze. Natomiast na zewnątrz prostokąta są miejsca ucieczki ciepła -- również o ustalonej, ale dużo niższej temperaturze. Nazwijmy je chłodnicami.

Symulacja odbywać się będzie krokowo. W każdym kroku komórka siatki oddaje ciepło tym sąsiadom, którzy mają niższą temperaturę niż ona. Utrata jest proporcjonalna do różnicy temperatur. Równocześnie komórka pobiera ciepło od tych sąsiadów, którzy mają wyższą temperaturę.

Mówiąc po prostu, zmiana wartości temperatury komórki jest proporcjonalna do sumy różnic temperatur z jej sąsiadami. Wielkość zmiany określa współczynnik proporcjonalności, jednakowy dla wszystkich komórek (czyli zakładamy jednakowe przewodnictwo cieplne wszystkich komórek). Niech *moja* oznacza aktualną temperaturę komórki, zaś *sąsiad_i* temperatury jej sąsiadów, wtedy

```
różnica := for i from 1 to 4 sum (sąsiadi - moja);
nowa-moja := moja + (różnica * współczynnik);
```

Dla komórek wewnętrznych za sąsiednie uznajemy 2 komórki po lewej i prawej stronie oraz 2 komórki powyżej i poniżej. Zewnętrzne komórki brzegowe (chłodnice) nie zmieniają temperatury (choć dostarczają ciepło lub je „kradną”). Na rysunku poniżej pokazano sąsiadów komórki oznaczonej gwiazdką.

	c	c	c	c	c	...	
c						...	c
c		g				...	c
c			S			...	c
c		S	*	S		...	c
c			S			...	c
c						...	c
c		g			g	...	c
c						...	c
...
	c	c	c	c	c	...	

Każdy krok to równoczesne policzenie przyrostów dla każdej komórki. Po obliczeniu przyrostów dla wszystkich komórek są one do nich dodawane i rozpoczynamy kolejny krok.

Część napisana w języku wewnętrznym powinna eksportować procedury wołane z C:

- `void start (int szer, int wys, float *M, float C, float waga)`

Przygotowuje symulację, np. inicjuje pomocnicze struktury. Argumentami są: rozmiary matrycy, początkowa zawartość matrycy (temperatury komórek), temperatura chłodnic oraz współczynnik proporcjonalności.

- `void place (int ile, int x[], int y[], float temp[])`

Umieszcza grzejniki w podanych miejscach i ustala ich temperatury.

- `void step ()`

Przeprowadza pojedynczy krok symulacji. Po jej wykonaniu macierz `M` (przekazana przez parametr procedury `start`) zawiera nowy stan.

Uwaga: parametr `M` jest deklarowany jako `float*` ze względów „poglądowych”, nie jest wymagane przekazanie tablicy `float`’ów. Można mieć dowolną reprezentację macierzy i użyć `cast`’a do przekazania jej adresu asemblerowi (on i tak nie zna typów). Być może elegantsze byłoby tu `void*`.

Dokładna postać wewnętrzna matrycy `M` nie jest określona (np. mogą to być dwie macierze), powinno być jednak możliwe jej łatwe zainicjowanie w programie w C przez wczytanie początkowej zawartości z pliku.

Testowy program główny napisany w C powinien zainicjować matrycę `M` oraz grzejniki `G` i chłodnice `C` (przez wczytanie ich zawartości z pliku). Nazwę pliku, współczynnik proporcjonalności i liczbę kroków symulacji podajemy jako argumenty wywołania programu z linii poleceń.

Po (prawie) każdym wywołaniu procedury `step()` powinno się wyświetlać aktualną sytuację, np. tekstowo, jako macierz liczb, po czym czekać na naciśnięcie <Enter>.

Postać danych na pliku

```
szerokość wysokość wartość-temperatury-dla-chłodnic
pierwszy wiersz M
....
ostatni wiersz M
liczba-grzejników
x, y i temp dla pierwszego grzejnika
....
x, y i temp dla ostatniego grzejnika
```

Mile widziany prosty plik testowy z danymi.

Rozwiązania nie zawierające pliku `Makefile` nie będą sprawdzane.

Rozwiązania (procedury w asemblerze i program w C z przykładowymi testami) należy wysłać do dnia 24 listopada (23:59) pocztą na `zbyszek@mimuw.edu.pl` jako **pojedynczy załącznik** -- archiwum o nazwie wskazującej na autora (np. `ab123456-zad2.tgz`), spakowane z osobnego katalogu o tej samej nazwie (ale bez `.tgz`). Program ma działać w środowisku zainstalowanym w laboratoriach w trybie 64-bitowym.