

5 Operacje na macierzach

5.1 Zadanie do wykonania w czasie ćwiczeń

5.1.1 Zadanie wstępne: Permutacje indeksów

Zadanie polega na sortowaniu łańcuchów znakowych bez zmiany ich położenia w pamięci, tj. każdy łańcuch pozostaje pod tym samym adresem. Zadanie obejmuje dwie wersje takiego sortowania – sortowanie tablicy wskaźników do łańcuchów oraz sortowanie łańcuchów zapisanych w dwuwymiarowej tablicy znakowej.

Łańcuchy są zapamiętywane w momencie definiowania tablicy wskaźników `keywords_tw[]`. Następnie są przepisywane do tablicy znakowej `keywords_t2D[]`. Przepisanie to realizuje funkcja `str_copy()`, której definicję należy dopisać do szablonu programu.

Główna część zadania polega na napisaniu definicji dwóch funkcji `sortuj_tw()` oraz `sortuj_t2D()`, które sortują w porządku alfabetycznym (np. metodą bąbelkową) łańcuchy – odpowiednio – wskazywane tablicą wskaźników albo zapamiętane w tablicy znakowej.

Istotnym warunkiem zadania jest pozostawienie łańcuchów w tym samym miejscu w pamięci, w jakim były przed sortowaniem. Funkcja `sortuj_t2D()` stosuje w tym celu wektor permutacji indeksów `indx`.

Po sortowaniu program wczytuje liczbę n i wywołuje funkcję `print_t2D()` (należy uzupełnić jej definicję), która ma wypisać w osobnych liniach n początkowych (alfabetycznie) łańcuchów.

- **Wejście**
0
liczba wypisywanych łańcuchów
- **Wyjście**
n początkowych łańcuchów
- **Przykład:**
Wejście:
0
3
Wyjście:
auto
break
case

5.1.2 Mnożenie macierzy

Szablon programu należy uzupełnić o definicję funkcji `mac_product()`, która oblicza iloczyn macierzy A i B i zapisuje go w macierzy AB .

- **Wejście**
1 liczba wierszy i liczba kolumn macierzy A
elementy macierzy A
liczba wierszy i liczba kolumn macierzy B
elementy macierzy B
- **Wyjście**
elementy macierzy AB
- **Przykład:**
Wejście: 1
2 3
1 2 3 10 20 30
3 2
11 23 1 1.5 -2 0
Wyjście: 7.0000 26.0000
70.0000 260.0000

5.1.3 Triangularyzacja macierzy i obliczanie wyznacznika - wersja uproszczona (bez zamiany wierszy)

W segmencie głównym programu są zdefiniowane tablice tablic (tablice „dwuwymiarowe”) $A[SIZE][SIZE]$, $B[SIZE][SIZE]$, $C[SIZE][SIZE]$, do których dane są wczytywane w `main`. Funkcja, której definicję należy uzupełnić, wykonuje obliczenia korzystając z tych tablic. Rozmiarów tych tablic nie należy zmieniać.

Szablon programu należy uzupełnić o definicję funkcji `gauss_simplified()`, która przekształca macierz kwadratową A do postaci trójkątnej górnej metodą Gaussa i zwraca wartość wyznacznika. W przypadku, gdy element na przekątnej głównej jest równy zeru, to triangularyzacja nie jest dokończona, a wyznacznik = NAN.

Funkcja może zmienić wartości elementów tablicy A.

- **Wejście**
2
liczba wierszy i macierzy A
elementy macierzy A
- **Wyjście**
wyznacznik macierzy
- **Przykład:**
Wejście: 2
3
1 2 3
9 8 7
1 0 0
Wyjście: -10.0000

5.2 Zadanie domowe

5.2.1 Rozwiązywanie układu równań liniowych metodą Gaussa - wersja z rozszerzaną macierzą współczynników

Szablon programu należy uzupełnić o definicję funkcji `gauss()`, która przekształca macierz kwadratową `A` do postaci trójkątnej górnej metodą Gaussa i zwraca wartość wyznacznika. Wiersze macierzy są zamieniane tak, aby wartość bezwzględna elementu głównego była największa. Zamiana wierszy nie jest realizowana poprzez przepisanie wierszy w tablicy, lecz z zastosowaniem wektora permutacji indeksów wierszy. W przypadku, gdy po zamianie wierszy element na przekątnej głównej jest mniejszy od `eps`, to triangularyzacja nie jest dokończana, a wyznacznik przyjmuje wartość 0.

Jeżeli argumenty funkcji `b` i `x` oraz wyznacznik nie są zerowe, to funkcja rozwiązuje układ równań i rozwiązanie zapisuje w tablicy `x`.

Funkcja może zmienić wartości elementów tablicy `A`.

Poprawność funkcji można sprawdzić korzystając z funkcji `mac_vec_product`

- **Wejście**

1
liczba wierszy i macierzy `A`
elementy macierzy `A`
elementy wektora `b`

- **Wyjście**

wyznacznik macierzy
elementy wektora `x`

- **Przykład:**

Wejście: 1
4
1 -1 2 -1
2 -2 3 -3
1 1 1 0
1 -1 4 3
-8 -20 -2 4

Wyjście: 4.0000
-7.0000 3.0000 2.0000 2.0000

5.2.2 Odwracanie macierzy kwadratowej metodą Gaussa - Jordana

Szablon programu należy uzupełnić o definicję funkcji `matrix_inv()`, która wyznacza (i zapamiętuje w tablicy `B`) macierz odwrotną do nieosobliwej macierzy zapisanej w tablicy `A`. Należy zastosować metodę Gaussa - Jordana z rozszerzaniem macierzy `A` o macierz jednostkową. Wiersze macierzy rozszerzonej są zamieniane analogicznie jak w zadaniu 3. Funkcja zwraca wyznacznik macierzy `A`. W przypadku, gdy po zamianie wierszy element na przekątnej głównej jest mniejszy od `eps`, to algorytm odwracania nie jest kończony, i wyprowadzany jest tylko wyznacznik

$= 0$ (układ równań nie jest rozwiązywany).

Funkcja może zmienić wartości elementów tablicy **A**.

Poprawność funkcji można sprawdzić korzystając z funkcji `mac_product`.

- **Wejście**

2

liczba wierszy i macierzy **A**

elementy macierzy **A**

- **Wyjście**

wyznacznik macierzy

elementy macierzy odwrotnej **B**

- **Przykład:**

Wejście: 2

3

1 2 -1

2 1 0

-1 1 2

Wyjście: -9.000

-0.2222 0.5556 -0.1111

0.4444 -0.1111 0.2222

-0.3333 0.3333 0.3333