Zadanie 1

(a) Zaimplementuj algorytmy realizujące podane poniżej problemy.
(b) Problemy należy rozwiązać **zgodnie** z przedstawionymi w pliku algorytmygrafowe.pdf instrukcjami. Uchroni to Państwa przed kopiowaniem zadania z Internetu.
(c) Komunikacja programu z użytkownikiem powinna się odbywać za pomocą plików o podanych poniżej formatach.

Problem 1 (najkrótsze ścieżki w grafie - algorytm Johnsona)

Na wejściu dany jest graf G = (V, E) skierowany, z wagami (zawierający zarówno wagi ujemne jak i dodatnie), bez cykli ujemnych reprezentowany tablicą list incydencji. Wyznacz długości najkrótszych ścieżek między wszystkimi parami jego wierzchołków wykorzystując przedstawiony na zajęciach algorytm Johnsona.

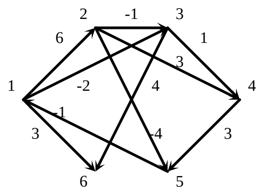
Dane:

- W pierwszej linii pliku In0401.txt znajduje się liczba naturalna n reprezentująca ilość wierzchołków grafu ważonego, skierowanego, bez cykli ujemnych G = (V, E).
- W kolejnych liniach pliku podano reprezentację grafu G w postaci tablicy list incydencji (w komentarzu przedstawiono, w jaki sposób należy interpretować kolejne wartości zapisane w pliku).

Wyjście:

- W pierwszej linii pliku Out0401.txt umieść wartości tablicy δ[] (reprezentujące długości najkrótszych ścieżek w grafie G' ze źródła s=0 do pozostałych wierzchołków grafu) wygenerowanej w wyniku realizacji algorytmu Forda-Bellmana.
- W kolejnych n+1 liniach podaj tablicę list incydencji (uwzględniając wagi krawędzi) reprezentującą graf G' = (V', E') z funkcją wagową $\hat{w}$: E'->R.
- W kolejnych n liniach wypisz wektory: $\hat{\delta}$ [s] (wektor długości najkrótszych ścieżek grafu G (z funkcją wagową $\hat{w}$: E->R) ze źródła s do wszystkich pozostałych wierzchołków grafu) oraz D[s] (wektor długości najkrótszych ścieżek grafu G (z funkcją wagową w: E->R) ze źródła s do wszystkich pozostałych wierzchołków grafu) dla s=1,...,n.
- Użyj formatu przedstawionego w przykładowym pliku Out0401.txt.

```
In0401.txt
6               // n
2 6 3 -2 6 3    // [1] 2(6) 3(-2) 6(3)
3 -1 4 3 5 -4   // [2] 3(-1) 4(3) 5(-4)
4 1 6 4         // [3] 4(1) 6(4)
5 3             // [4] 5(3)
1 -1            // [5] 1(-1)
                // [6]
```




```
Out0401.txt
0 –5 0 –7 –6 –4 –3
[0]  1(5) 2(0) 3(7) 4(6) 5(4) 6(3)
[1]  2(1) 3(0) 6(5)
[2]  3(6) 4(9) 5(0)
[3]  4(0) 6(0)
[4]  5(1)
[5]  1(0)
[6]
Delta^[1][0 1 0 0 1 0], D[1][0 6 -2 -1 2 2]
Delta^[2][0 0 0 0 0 0], D[2][-5 0 -7 -6 -4 -3]
Delta^[3][1 2 0 0 1 0], D[3][3 9 0 1 4 4]
Delta^[4][1 2 1 0 1 1], D[4][2 8 0 0 3 4]
Delta^[5][0 1 0 0 0 0], D[5][-1 5 -3 -2 0 1]
Delta^[6][∞ ∞ ∞ ∞ ∞ 0], D[6][∞ ∞ ∞ ∞ ∞ 0]
```

Problem 2 (maksymalny przepływ w sieci – algorytm Forda-Fulkersona)

Dana jest sieć G z wyróżnionym źródłem s i ujściem t. Wyznacz maksymalny przepływ w sieci G metodą Forda-Fulkersona (z użyciem stosu).

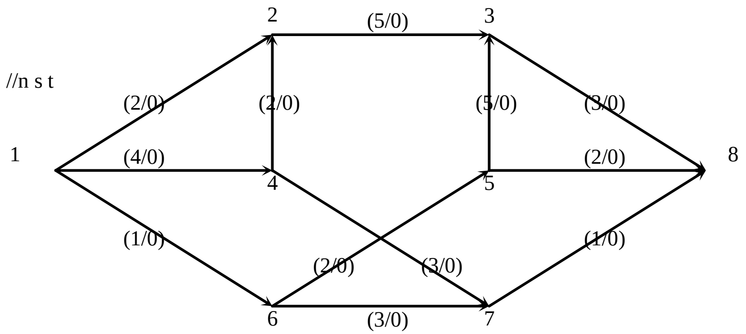
Dane:

- W pierwszej linii pliku In0402.txt znajduje się liczba naturalna n (reprezentująca ilość węzłów sieci G=(V, E)), liczba s (reprezentująca źródło) i liczba t (reprezentująca ujście).
- W kolejnych liniach pliku podano reprezentację sieci G w postaci tablicy list incydencji. Drugi wiersz przykładowego pliku należy interpretować w następujący sposób "**2** 2 **4** 4 **6** 1". W grafie G istnie krawędź: (1, **2**), (1, **4**) i (1, **6**). Krawędź (1, **2**) ma przepustowość 2, krawędź (1, **4**) ma przepustowość 4 i krawędź (1, **6**) ma przepustowość 1.

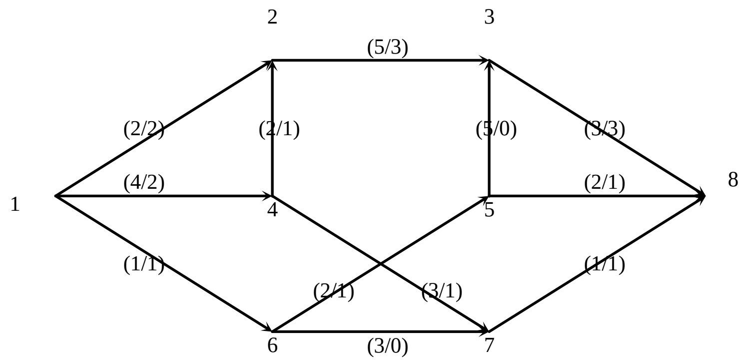
Wyjście:

- W kolejnych k liniach pliku Out0402.txt znajduje się na przemian: zawartość stosu ustalonego przy generowaniu kolejnej ścieżki i sama ścieżka (zapisana od ujścia do źródła).
- W kolejnych n liniach znajduje się tablica list incydencji sieci (w nawiasach należy podać wyznaczone przepływy krawędzi)
- W ostatniej linii pliku należy podać wartość przepływu ujścia t.

```
Przykład
In0402.txt
8 1 8           //n s t
2 2 4 4 6 1
3 5
8 3
2 2 7 3
3 5 8 2
5 2 7 3
8 1
```




```
Out0402.txt
STOS: 1 2 4 6 5 7
8 7 6 1
STOS: 1 2 4 7 6 5 3
8 5 6 7 4 1
STOS: 1 2 4 7 3
8 3 2 1
STOS: 1 4 2 7 3
8 3 2 4 1
STOS: 1 4 2 7 3
[1] 2 2, 4 2, 6 1
[2] 3 3
[3] 8 3
[4] 2 1, 7 1
[5] 3 0, 8 1
[6] 5 1, 7 0
[7] 8 1
[8]
3+1+1=5
```



Zestawy

Zestaw	1	2
Zad1	1	2
