

PPJ – zadania do zaprogramowania

1. *Generowanie i prezentacja tablic 2 wymiarowych.* W różnych zastosowaniach, w których mamy do czynienia z tablicami prostokątnymi, trzeba prezentować wyniki tabelaryczne zachowując wyrównanie kolumn wg najbardziej „wymagającego”, czyli największego co do liczby znaków, elementu w kolumnie. Na przykład, dla tablicy 3 x 4 zawierającej elementy podane z lewej strony powinniśmy otrzymać wydruk z prawej (wyrównanie w kolumnach prawostronne):

```
7 -123 1 257      7 -123 1 257
1234 555 3 15    ==> 1234 555 3 15
4 -15 0 0         4 -15 0 0
```

Napisać program, który wg podanych przez użytkownika 3 parametrów m , n , z tworzy tablicę m wierszy na n kolumn wypełnioną elementami generowanymi losowo z zakresu $-z .. z$. Tablica jest pokazywana w dwu wersjach: bez wyrównania kolumn (pojedyncze odstępy pomiędzy elementami w wierszu) i z wyrównaniem. **Wskazówka:** Można obliczać szerokość pola wydruku liczby posługując się metodą `printf()`. Oto przykład:

```
// Sterowanie szerokością pól w printf()
int w1=3, w2=5, w3 = 7; // Szerokości pól
System.out.printf("12345678901234567890\n");
System.out.printf("%"+w1+"d%"+w2+"d%"+w3+"d\n", 11, 22, 33);
```

Wynik:

```
12345678901234567890
11    22    33
```

2. *Tablica trójkątna.* Zadanie podobne do 1. Wydrukować generowaną losowo macierz poddiagonalną $n \times n$. W wierszu 1 tylko 1 element, w wierszu 2 – dwa elementy, ..., w wierszu n – n elementów. W tym ćwiczeniu szerokość pola wydruku liczb całkowitych jest stała dla całej tablicy.
3. *Wielkie liczby – BigInteger.* Napisać program, który dla zadanej przez użytkownika liczby n z przedziału 1..99 oblicza liczbę cyfr rozwinięcia dziesiętnego n^n . Opcjonalnie pokazać także wartość n^n w zapisie dziesiętnym i heksadecymalnym. Podobnie zaprezentować wartość $n!$.
4. *Bilans nawiasów.* Napisać program, który pobiera tekst ze wskazanego przez użytkownika pliku tekstowego i sprawdza poprawność struktur nawiasowych `()`, `[]`, `{}`. W szczególności analiza poprawnego składniowo pliku źródłowego Java powinna dać wynik pozytywny.
Uwaga: problemem w odniesieniu do plików .java jest obecność nawiasów w komentarzach i łańcuchach znaków, w których bilansowanie nie obowiązuje. Dla uproszczenia proszę nie zajmować się neutralizacją komentarzy i łańcuchów.
5. *Skanowanie tablicy.* Napisać program wypełniający tabelę m wierszy na n kolumn wg wybranego schematu skanowania podanego w tabelce poniżej. Tablica powinna być wypełniona liczbami 1 .. $m \cdot n$ w kolejności odwiedzania elementów.

LGw, LDw, PGw, PDw LGk, LDk, PGk, PDK	LG – start w narożniku lewym-górnym, LD – lewym-dolnym PG – prawym-górnym, PD – prawym-dolnym. w – skanowanie wierszami, k – skanowanie kolumnami.
--	--

Na przykład tablica 3 X 5 wypełniona wg schematu skanowania PDw ma postać:

```
15 14 13 12 11
 6  7  8  9 10
 5  4  3  2  1
```

6. *Praca z plikami znakowymi przy pomocy Scanner'a i PrintWriter'a:* Napisać program, który wczytuje z pliku tekstowego wskazanego przez użytkownika tablicę prostokątną liczb całkowitych, porządkuje elementy w wierszach rosnąco, a następnie wyprowadza tak uporządkowaną tablicę do pliku wyjściowego (można skorzystać z drukowania wyrównującego

kolumny). W pliku wejściowym może być wiele tablic – trzeba je wszystkie przetworzyć zgodnie z podanym scenariuszem. Format zapisu danych w pliku wejściowym (liczby oddzielone „białymi plamami”):

m n
A1 A2 ... Amn [łamanie wierszy dowolne]

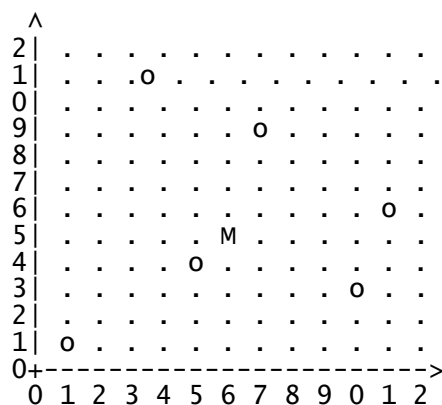
Przykład: Jeżeli plik **dane.txt** zawiera:

```
3 3
1111 -92 0 7 5 1234 37 -10 -23
1 1 9999
```

to wynik wysłany do pliku wyjściowego (powiedzmy **wynikiDla_dane.txt**) powinien zawierać:

```
3 3
-92 0 1111
5 7 1234
-23 -10 37
1 1
9999
```

7. *Punkty 2D.* Zdefiniować klasę do reprezentowania punktów na płaszczyźnie (wg współrzędnych kartezyjańskich całkowitych) i użyć jej do generowania losowego zbioru punktów w pierwszej ćwiartce płaszczyzny, na podstawie parametrów dostarczonych przez użytkownika: n , $zakres$. Dla zbioru punktów wyznaczyć punkt medianowy M , to znaczy taki punkt, którego współrzędna $M.x$ jest medianą współrzędnych x punktów zbioru i podobnie – współrzędna $M.y$ jest medianą współrzędnych y punktów zbioru. Dla małych konfiguracji zaprezentować zbiór punktów znakowo („pseudograficznie”), na przykład stosując konwencję podaną poniżej (6 punktów w zakresie współrzędnych 0..12)



8. *Układ równań liniowych.* Napisać program(ik) rozwiązujący układ 2 równań liniowych o współczynnikach całkowitych. Wykorzystać klasę **Ulam** do przeprowadzenia obliczeń. Na przykład jeżeli użytkownik poda współczynniki i wyrazy wolne:

```
3 5 7
1 4 2
```

to program powinien znaleźć rozwiązania: $x = 18/7$, $y = -1/7$

9. *Kwadraty magiczne.* Napisać program generujący kwadrat magiczny rzędu n dla zadanego przez użytkownika parametru k , $n = 2k+1$. Tak zwana *metoda syjamska* wypełniania kwadratu polega na posuwaniu się w tablicy przekątniowo w prawo w górę przy cyklicznym indeksowaniu (tablica „zwinęta” w torus); jeżeli następny element jest już wypełniony, to wybieramy pierwszy wolny w kolumnie poniżej elementu aktualnego. Start w środku wiersza 0.