
Zajęcia 3 – tablice

1. Napisać program, który utworzy tablicę 10 liczb całkowitych i wypełni ją wartościami losowymi z przedziału $[-10, \dots, 10]$, a następnie:
 - a) wypisze na ekranie zawartość tablicy,
 - b) wyznaczy najmniejszy oraz największy element w tablicy,
 - c) wyznaczy średnią arytmetyczną elementów tablicy,
 - d) wyznaczy ile elementów jest mniejszych, ile większych od średniej.
 - e) wypisze na ekranie zawartość tablicy w odwrotnej kolejności, tj. od ostatniego do pierwszego.

Wszystkie wyznaczone wartości powinny zostać wyświetlone na ekranie.

Wylosowane liczby:

-3 9 2 -10 -3 -4 -1 -5 -10 8

Min: -10, max: 9

Średnia: -1,00

Mniejszych od śr.: 6

Większych od śr.: 3

Liczby w odwrotnej kolejności:

8 -10 -5 -1 -4 -3 -10 2 9 -3

2. Napisać program, który utworzy tablicę 20 liczb całkowitych z przedziału $1 \dots 10$, a następnie wypisze na ekranie ile razy każda z liczb z tego przedziału powtarza się w tablicy.

Przykład:

Wylosowane liczby: 6 5 4 5 10 5 8 3 10 6 6 6 4 3 2 8 1 3 4 7

Wystąpienia:

1 - 1

2 - 1

3 - 3

4 - 3

5 - 3

6 - 4

7 - 1

8 - 2
9 - 0
10 - 2

3. Napisz program, który stworzy tablicę `m` (macierz) 5 x 5 liczb całkowitych i wypełnij ją losowymi wartościami z zakresu $\{-5, -4, \dots, 5\}$, a następnie:
- wyświetli na ekranie wylosowaną macierz,
 - wyświetli minimalną wartość w każdym wierszu,
 - wyświetli maksymalną wartość w każdej kolumnie,
 - wyświetli maksimum dla 1. przekątnej, czyli od komórki `m[0][0]` do komórki `m[4][4]`,
 - wyświetli minimum dla 2. przekątnej, czyli od komórki `m[0][4]` do komórki `m[4][0]`.

Przykład.

Wylosowana macierz:

```
 2  1 -2 -5 -2
 3  0 -1  1 -4
-5 -1  3  1  4
-2 -2  1  1  2
-2 -4  4 -5  2
```

Minimalne wartości w wierszach:

```
0: -5
1: -4
2: -5
3: -2
4: -5
```

Maksymalne wartości w kolumnach:

```
0:  3
1:  1
2:  4
3:  1
4:  4
```

Maksimum na 1. przekątnej: 3

Minimum na 2. przekątnej: -2

4. Napisać program, który wczytuje od użytkownika liczbę całkowitą, a następnie wyświetla jej reprezentację w kodzie binarnym (ZM). Podczas konwersji liczby należy kolejne jej bity zapisywać w pomocniczej tablicy liczb całk. o rozmiarze 32. Konwersji należy dokonać korzystając z operacji dzielenia całkowitego oraz operacji modulo.

Przykład:

Wejście:

-75 (liczba podana przez użytkownika)

Wynik:

Liczba -75 binarnie: 1.1001011

5. Napisać program na podstawie programu do zadania 4, który dodatkowo wyświetli liczbę w pozostałych kodach, a więc ZU1 oraz ZU2.
6. Napisz program, który pobiera od użytkownika dodatnią liczbę naturalną n i tworzy tablicę a zmiennych typu logicznego (boolean) o rozmiarze $n \times n$. Następnie program powinien wypełnić utworzoną tablicę, tak by $a[i][j] = \text{true}$ jeżeli liczby $(i+1)$ oraz $(j+1)$ są względnie pierwsze, tzn. nie mają wspólnych dzielników poza 1. Tak utworzoną tablicę należy wypisać na ekranie, przy czym dla wartości `true` należy wyświetlić znak "+", natomiast dla wartości `false` znak ".". Przykład:

Podaj liczbę (> 0): 10

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | + | + | + | + | + | + | + | + | + | + |
| 2 | + | . | + | . | + | . | + | . | + | . |
| 3 | + | + | . | + | + | . | + | + | . | + |
| 4 | + | . | + | . | + | . | + | . | + | . |
| 5 | + | + | + | + | . | + | + | + | + | . |
| 6 | + | . | . | . | + | . | + | . | . | . |
| 7 | + | + | + | + | + | + | . | + | + | + |
| 8 | + | . | + | . | + | . | + | . | + | . |
| 9 | + | + | . | + | + | . | + | + | . | + |
| 10 | + | . | + | . | . | . | + | . | + | . |

7. Gra w życie (The game of life) to jeden z najbardziej znanych przykładów automatu komórkowego, wymyślony w roku 1970 przez brytyjskiego matematyka Johna Hortona Conwaya. Gra toczy się na nieskończonej planszy podzielonej na komórki. Każda komórka ma ośmiu sąsiadów (po bokach i po skosie). Każda komórka może być w jednym z dwóch stanów: *żywa* albo *martwa*. Bieżący stan komórek planszy używany jest do wyznaczenia kolejnego stanu komórek na planszy (w następnej jednostce czasu)

Stan komórek ulega zmianie zgodnie z następującymi regułami:

- martwa komórka, która ma dokładnie 3 żywych sąsiadów, staje się żywa w następnej jednostce czasu (rodzi się),
- żywa komórka z 2 albo 3 żywymi sąsiadami pozostaje żywa; przy innej liczbie sąsiadów umiera (z „samotności” albo „zatłoczenia”).

Napisać program ilustrujący działanie opisanego automatu w kolejnych momentach czasu. Do przechowywania *bieżącego* i *kolejnego* stanu planszy należy użyć dwóch tablic liczb całkowitych o wymiarach $n \times n$ (0 – komórka martwa, 1 – żywa). W celu zapewnienia tej samej liczby sąsiadów dla każdej komórki zakładamy, że komórki *brzegowe* z jednej strony planszy sąsiadują z komórkami po drugiej stronie planszy, tj. lewym sąsiadem komórki o indeksie $[w][0]$ jest komórka $[w][n - 1]$, dolnym sąsiadem komórki o indeksie $[n - 1][k]$ jest komórka $[0][k]$, itd. Program powinien wyświetlać kolejne stany planszy (komórkę żywą oznaczyć krzyżykiem #, natomiast martwą kropką).

Przykład działania dla 4 iteracji dla planszy 6×6 oraz komórek tworzących tzw. *szybowiec*, który to jest wzorcem *wiecznie żywym*, chociaż zmieniającym kształt i położenie na planszy.

Iteracja 0

```
.....
....#.
.....#
...###
.....
```

```
.....
.....
```

Iteracja 1

```
.....
.....
...#.#
....##
....#.
.....
```

Iteracja 2

```
.....
.....
.....#
...#.#
....##
.....
```

Iteracja 3

```
.....
.....
....#.
#....#
....##
.....
```

8. Napisać program na podstawie programu do zadania 4 i 5, który dokonuje mnożenia liczb II metodą Booth'a.