Laboratorium 3B (7pkt)

Celem programu jest prezentacja operacji związanych ze statycznymi tablicami dwuwymiarowymi.

Przygotowany jest szkielet programu wraz z funkcją **main**. Każdy etap w funkcji main należy wykonać i sprawdzić jego działanie poprzez odkomentowanie kolejnych sekcji. Wszystkie etapy należy wykonać w podanej kolejności. Funkcje należy uzupełnić, zarówno prototypy (na podstawie wywołań funkcji w main) jak i ciała funkcji. W funkcji main **nie należy nic zmieniać**, poza komentowaniem/odkomentowaniem oraz dodaniem deklaracji odpowiednich zmiennych. Przygotowana jest funkcja **animation**, której także nie należy modyfikować. Jest odpowiedzialna za animację tworzonych figur. Jeśli poszczególne metody będą dobrze zrobione, animacja będzie działać. Gotowa jest także funkcja **bottomTriangle** – też nie modyfikujemy.

We wszystkich funkcjach posługujemy się tablicami statycznymi. W żadnej funkcji nie wolno też używać tablic pomocniczych, wszystkie operacje mają się odbywać w oryginalnej tablicy.

Kolejne etapy zadania to napisanie funkcji przedstawionych poniżej:

```
void fillTable (/*...*/);
void setInRow(/*...*/);
void printTable (/*...*/);
void christmasTree (/*...*/);
void rectangle(/*...*/);
void rightTriangle (/*...*/);
void transposition (/*...*/);
```

Pierwszym argumentem wejściowym jest zawsze tablica, kolejne to jej wymiary i na końcu inne zmienne.
Do tablicy będziemy wpisywać albo zera, albo jedynki.

Etap 1 (0 pkt)
W zerowym etapie przedstawione jest działanie funkcji zmieniającej kolor czcionki w konsoli. Będzie to potrzebne w następnym punkcie. Tu nic nie trzeba robić, tylko się zapoznać.

Etap 1 (2,5 pkt)
Funkcja `fillTable` wypełnia całą tablicę wartością podaną w czwartym argumencie. Funkcja `setInRow` zmienia tablicę, wpisując w podany wiersz (czwarty argument) od początku wiersza kilka razy jedną liczbę, potem kilka razy drugą, reszta wiersza pozostaje bez zmian. Funkcja `printTable` wypisuje tablicę: kolejne wiersze jeden pod drugim, elementy w wierszach bez odstępu. Ponadto wypisuje zera kolorem czerwonym, natomiast inne wartości kolorem domyślnym. W przykładzie poniżej w wierszu o indeksie 5 wpisane są 3 dwójki, potem 4 zera.

```
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
22200001111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
```

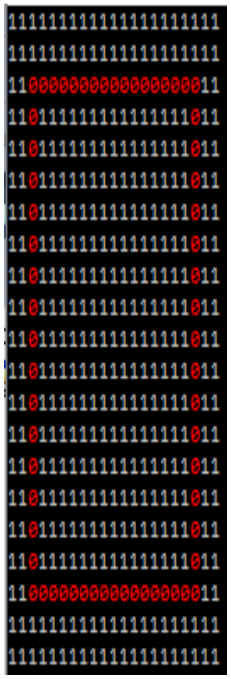
Laboratorium 3B (7pkt)

Etap 2 (1 pkt)

Funkcja `christmasTree` w tablicy wejściowej rysuje choinkę. W kolejne wiersze wpisuje jedynki i zera. Ilość jedynek w kolejnych wierszach podana jest w drugiej kolumnie tablicy o nazwie **tree**, ilość zer w trzeciej. Pierwsza kolumna tej tablicy określa nr wiersza, druga ilość wstawianych jedynek, trzecia ilość zer. W implementacji tej funkcji musi być użyta funkcja `setInRow`.

Etap 3 (1.5 pkt)

Funkcja `rectangle`. W tablicy wejściowej wpisuje niewypełniony prostokąt za pomocą zer. Czwarty argument określa indeks wiersza i kolumny, w którym jest lewy górny róg prostokąta. Przykład: indeks = 2, boki prostokąta w trzecim wierszu, trzeciej kolumnie, trzecim wierszu od końca i trzeciej kolumnie od końca.



Etap 4 (2,0 pkt)

Funkcja `rightTriangle`. Wpisuje do tablicy wypełniony prawy trójkąt, za pomocą zer. Czwarty argument określa indeks od której kolumny zaczyna się bok trójkąta (licząc od zera i od końca). Wierzchołek trójkąta ma dochodzić do środka tablicy. Wypełniamy tylko prawą połową tablicy, lewą pozostawiając niezmienioną. Przykład: indeks kolumny = 1.

Funkcja `transposition`. Zamienia w tablicy wiersze z kolumnami. Zakładamy, że tablica jest kwadratowa.

Laboratorium 3B (7pkt)