Laboratorium – 01.06.2021

1. (**40 pkt**) Napisz funkcją *dec2bin*, która będzie zwracała reprezentację binarną liczby całkowitej zapisanej w systemie dziesiętnym:
   a) Funkcja **zwraca wskaźnik to tablicy**, której elementami są cyfry reprezentacji binarnej liczby całkowitej. **Pierwszym elementem tej tablicy jest rozmiar tablicy**
   b) Funkcja przyjmuje jeden argument w postaci **liczby całkowitej**
   c) Rozmiar tablicy, potrzebnej do przechowania zapisu dziesiętnego, znajdujemy, obliczając logarytm o podstawie 2 liczby w zapisie dziesiętnym, rzutując wynik na *int* oraz dodając 1. **Pamiętajmy, że rozmiar tablicy musi zostać dodatkowo zwiększony o 1, żeby przechować informację o rozmiarze tablicy.**
   d) W funkcji tworzymy tablicę, używając funkcji *calloc*, zdefiniowanej w bibliotece *stdlib.h*
   e) Algorytm poszukiwania rozwinięcia dziesiętnego – zakładamy, że liczba ma *K* cyfr w zapisie dziesiętnym:
      - Tworzymy pętlę for o indeksie *i* od 0 do indeksu mniejszego niż *K*
      - Sprawdzamy warunkiem *if* czy *liczba*<2^{K-1-i}. Jeśli tak, to do komórki *i*-tej wpisujemy 0. W przeciwnym razie do komórki wpisujemy 1, a od liczby odejmujemy 2^{K-1-i}

W main:

a) Pobierz liczbę całkowitą, której reprezentację szukamy, funkcją *scanf*
b) Wywołaj funkcję dec2bin dla pobranej liczby, a wartość zwracaną przez funkcję przypisz wskaźnika, który zadeklaruj wcześniej
c) Wypisz reprezentację binarną liczby do konsoli wywołując *printf*: **Reprezentacja binarna liczby <*liczba_przekazana*> jest równa <*wypisanie tablicy*>**. Żeby wypisać elementy tablicy wykorzystaj pętlę for oraz wiedzę, że rozmiar tablicy mieści się w pierwszym elemencie wypisywanej tablicy
d) Pamiętaj o zwolnieniu pamięci funkcją *free*

Przykład: Poszukajmy reprezentacji liczby 10:
Rozmiar tablicy *tab* wynosi: *int log(10)/log(2)+1=4* – potrzebujemy tablicy o 4 komórkach, żeby zapisać liczbę 10 w systemie binarnym. Do tego rozmiaru dodajemy 1, ponieważ chcemy przechować informację o rozmiarze tablicy. Ostateczny rozmiar tablicy to *N*=5. Zwrócona tablica będzie miała postać*: tab*=[51010].

Uwaga! Należy uważać na rozmiar tablicy. Rozmiar tablicy w tym wypadku jest o jeden większy niż maksymalna potęga liczby 2. W algorytmie wyznaczania liczby binarnej musimy używać wyrażenia 2^{K-2}.

2. (**10 pkt**)Stwórz funkcję *create_mat*, która stworzy dynamiczną macierz 2D o rozmiarze $N \times M$:
   a) Funkcja zwraca wskaźnik do tablicy 2D
   b) Funkcja przyjmuje trzy argumenty, z których pierwszy inicjuje liczbę wierzy, a drugi liczbę kolumn macierzy. Trzeci argument jest parametrem, który informuje czy macierz będzie macierzą z elementami równymi 0 czy wypełnioną losowymi liczbami z przedziału <0, 1>

c) Jeśli parametr jest równy 0, to elementy macierzy są równe 0, w przeciwnym wypadku macierz wypełniamy losowymi liczbami.

3. (**5 pkt**) Napisz funkcję *free_mat,* która zwolni pamięć tablicy 2D, której wskaźnik przekażemy do argumentu funkcji. Do argumentu tej funkcji przekazujemy również liczbę wierszy tablicy.
   Uwaga! Najpierw zwalniamy pamięć, przydzieloną do każdego z wierzy tablicy. Na koniec zwalniamy wskaźnik tablicy 2D.
4. (**5 pkt**) Napisz funkcję *print_mat*, która wypisze tablicę 2D do konsoli. Argumentami funkcji jest wskaźnik do wskaźnika typu *double* oraz liczba wierzy i kolumn tablicy. Tablicę wypisujemy wiersz po wierszu.
5. (**40 pkt**) Stwórz funkcję *mult_mat*, która pomnoży przez siebie dwie macierze oraz zwróci wskaźnik do tablicy dwuwymiarowej.
   a) Funkcja zwraca wskaźnik to wskaźnika do liczby typu *double*
   b) Funkcja przyjmuje jako argument dwie zmienne, będące wskaźnikiem do wskaźnika typu *double* oraz 3 zmienne typu *int*: liczbę wierszy pierwszej macierzy, liczbę kolumn pierwszej macierzy oraz liczbę kolumn drugiej macierzy. **Nie trzeba przekazywać liczby wierszy drugiej macierzy, bo z definicji mnożenia macierzowego wynika, że musi być ona równa liczbie kolumn pierwszej macierzy.**
   c) Jeśli rozmiar macierzy pierwszej to $N_1 \times M_1$, natomiast rozmiar macierzy drugiej to $M_1 \times M_2$, to iloczyn macierzy pierwszej i macierzy drugiej ma rozmiar $N_1 \times M_2$, - proszę utworzyć w funkcji dynamiczną macierz 2D, przechowującą wynik mnożenia macierzonego, o takich rozmiarach. Proszę użyć do tego funkcji *create_mat* z parametrem 0.

   W main:
   a) Zadeklaruj dynamiczną macierz pierwszą o rozmiarze 5×6. Wypełnij ją losowymi liczbami z przedziału <0, 1>
   b) Zadeklaruj druga dynamiczną macierz o rozmiarze 6×4. Wypełnij ją losowymi liczbami z przedziału <0, 1>.
   c) Wypisz obie tablice do konsoli.
   d) Wywołaj dla obu utworzonych tablic funkcję *mult_mat*
   e) Wypisz wynik działania funkcji *mult_mat* do konsoli
   f) Zwolnij pamięć przydzieloną dla każdej z macierzy.

Przypomnienie! Niech Macierz $\boldsymbol{C_{N\times M_2} = A_{N\times M_1} B_{M_1\times M_2}}$. Element macierzowy $\boldsymbol{C[i,j] = \sum_{k=0}^{M_1-1} A[i,k] \cdot B[k,j]}$ – indeksacja elementów przebiega od 0, tj. $\boldsymbol{i = [0, 1, \dots, N-1]}$, $\boldsymbol{j = [0, 1, \dots, M_2 - 1]}$