Assignment #3: Matrix calculations in modern IEEE standards

Note: It is prohibited to use any of the following functions in your implementation: `horzcat()`, `vertcat()`, `transpose()`, as well as any form of cycle (`for`, `while`) and `if` conditions. You are required to use MATLAB operators instead. Do not modify existing function prototypes. Submit only those exercises that you have ready and that you've already tested. Programs that crash because of a parsing error, as well as commented-out code, will not be evaluated.

Exercise 1: Vector cyclic shift function.

Implement the body of the function `rotateRight( M, shift)`, that performs cyclic shift of vector's position - the ROR operation on a row-vector. If a matrix is supplied as the first input parameter, rotate all rows. The scalar `shift` parameter specifies how many positions to rotate. If negative, a ROL operation should performed. Do not use any loops. **This exercise is to be implemented using vector(matrix) indexing. The use of any form of cycles, or other toolbox functions, such as `circshift()`, is explicitly prohibited.**

Exercise 2: Circular shift matrix

Implement the body of the function `cyclicShiftMatrix( Size, Shift )`, that generates a square permutation matrix of a given size. The permutation is a right rotation - the ROR operation. That means the right-multiplication of a row vector by this matrix performs a cyclic shift of vector's position - the ROR operation. The scalar `Shift` parameter specifies how many positions to rotate. If negative, a ROL operation should be performed. **Do not use any loops in your implementation. This exercise is to be implemented using the working implementation of the function `rotateRight()`, from Exercise 1. The use of other toolbox functions, such as `circshift()`, is explicitly prohibited.**

Exercise 3: Implement model matrix expansion

Implement (finish the implementation) the body of the function `WIFI6_LDPC_ExpandH( Hbm, z )`, that expands the integer model matrix $\mathbf{H}_{bm}$ to a large binary sparse matrix **H**, as was explained in detail during the course lectures. You may also use the provided excerpts from the WiFi-6: IEEE 802.11-2020 and IEEE 802.16-2017 standards as supplementary documentation. **This exercise is to be implemented using the working implementation of the function `cyclicShiftMatrix()`, from the Exercise 2. The use of `for` loops and `if` conditions is permitted only for iterating over the elements of $\mathbf{H}_{bm}$ (and submatrices of H).**

Exercise 4: Implement direct encoding using the H matrix submatrices

Implement (finish the implementation) the body of the function `encodeLDPC( U, H )`, that encodes a binary vector **u** (denoted also as **i**) of length *k* bits to a binary vector **c** of length *n* bits using the procedure described in the lecture and also by the provided excerpt from the IEEE 802.16-2017 standard Annex G.3 Method 2. As optional parameters the function is required to return the A, B, T, C, D, E and T^{-1} submatrices.

Zadanie #3: Použitie matíc v moderných štandardoch IEEE

Poznámka: Pri implementácii je zakázané použitie funkcií: `horzcat()`, `vertcat()`, `transpose()` ako aj všetkých foriem cyklov (napr. `for` a `while`) a podmienok `if`. Miesto nich je povinné použitie operátorov jazyka MATLAB. Nemodifikujte existujúce funkčné prototypy. Odovzdávajte len tie úlohy, ktoré máte skutočne pripravené a otestované. Programy, ktoré ani nepôjdu spustiť v dôsledku syntaktickej chyby, ako aj zakomentovaný kód, nebudú hodnotené vôbec.

Úloha 1: Cyklický posun vektora

Implementujte telo funkcie `rotateRight( M, shift )`, ktorá vykoná cyklický posun vstupného riadkového vektora `M` vpravo (operáciu ROR). Ak je prvým vstupným parametrom funkcie matica, vykonajte tento posun naraz pre všetky jej riadky. Skalárny parameter `shift` špecifikuje, o koľko pozícií sa má vektor/matica posunúť. Ak je zadaná negatívna hodnota, funkcia má implementovať cyklický posun vľavo (operáciu ROL). **Pri implementácií je zakázané používať akékoľvek cykly. Predpísané je využitie vektorovej/maticovej indexácie jazyka MATLAB. Použitie funkcie `circshift()`, alebo podobných, je explicitne zakázané.**

Úloha 2: Matica cyklického posuvu

Implementujte telo funkcie `cyclicShiftMatrix( Size, Shift )`, ktorá vygeneruje štvorcovú permutačnú maticu zadaného rozmeru. Nech je permutácia cyklický posun – operácia ROR. Teda nech ak riadkový vektor násobí túto maticu z ľava, potom výsledný riadkový vektor predstavuje jeho posun o `Shift` pozícií doprava. Ak je parameter `Shift` záporný, násobenie vygenerovanou maticou má implementovať cyklický posun vektora doľava. **Pri implementácií je predpísané využitie existujúcej funkcie `rotateRight()` z predchádzajúcej úlohy. Použitie akýchkoľvek cyklov, alebo iných funkcií toolboxov je explicitne zakázané.**

Úloha 3: Implementácia expanzie modelovej matice $\mathbf{H}_{bm}$ na maticu H

Implementujte (resp. dokončite dodanú kostru implementácie) funkcie `WIFI6_LDPC_ExpandH( Hbm, z )`, ktorá vykoná expanziu celočíselnej modelovej matice $\mathbf{H}_{bm}$ na veľkú binárnu riedku maticu **H**, podľa postupu opísaného na prednáške. Použite dodané výseky zo štandardu pre WiFi-6: IEEE 802.11-2020 a IEEE 802.16-2017 môžete použiť ako doplňujúcu dokumentáciu. Pri implementácií je predpísané využitie funkcie **`cyclicShiftMatrix()` z predchádzajúcej úlohy. V tejto úlohe je povolené použitie cyklov `for` a podmienok `if` a to len pre iteráciu cez prvky $\mathbf{H}_{bm}$ (resp. submatice H).**

Úloha 4: Implementácia priameho kódovania pomocou submatíc matice H

Dokončite implementáciu funkcie `encodeLDPC( U, H )`, ktorá zakóduje binárny vektor **u** (označovaný aj ako **i**) dĺžky *k* bitov na binárny vektor **c** dĺžky *n* bitov podľa postupu popísaného na prednáške a vo výseku zo štandardu IEEE 802.16-2017 Annex G.3 Method 2.Ako nepovinné výstupné parametre musí funkcia vedieť vracať submatice A, B, T, C, D, E aj T^{-1}.