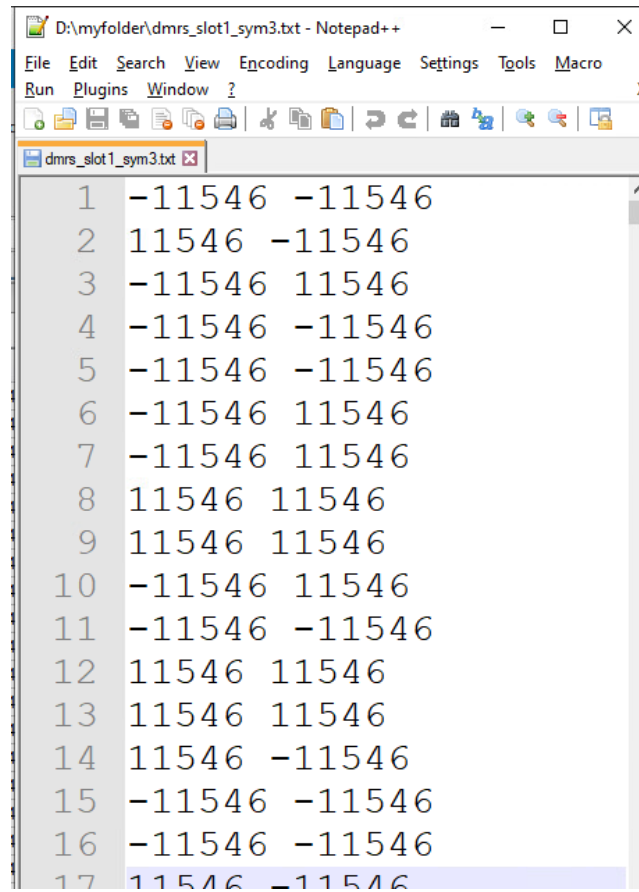


BÀI TẬP ÔN TẬP LẬP TRÌNH C CHO SINH VIÊN THỰC TẬP PHÒNG XLTH

Tuần	Task	Output	Start Date	End Date
1	Bài tập 1. Viết chương trình C thực hiện: <ul style="list-style-type: none"> - Nhập 2 chuỗi (strings) và 1 ký tự từ bàn phím. - Sử dụng con trỏ để ghép nối 2 strings, so sánh 2 strings, in ra màn hình 2 strings theo thứ tự đảo ngược, chuyển các chữ thường trong chuỗi thành chữ viết hoa, kiểm tra xem ký tự đã nhập có xuất hiện trong 2 chuỗi hay không, nếu có đưa ra số lần xuất hiện. 	Mã nguồn C, với các file input, output theo yêu cầu. Code yêu cầu viết rõ ràng, có chú thích cụ thể.	2/4	6/4
2	Bài tập 2: Viết chương trình C thực hiện: <ul style="list-style-type: none"> - Nhập số nguyên dương N ($N \leq 200$) từ bàn phím - Tạo ngẫu nhiên mảng các số nguyên (-32768 to 32767) gồm N phần tử - Sắp xếp mảng theo thứ tự tăng dần và loại bỏ các phần tử trùng nhau (nếu có) - Chuyển đổi mảng các số nguyên sang hexa và binary. - Lưu các kết quả ra file, bao gồm: N, mảng số nguyên trước và sau sắp xếp, mảng số hexa, mảng số binary. 	Mã nguồn C, với các file input, output theo yêu cầu. Code yêu cầu viết rõ ràng, có chú thích cụ thể.		
3	Bài tập 3: Viết chương trình C thực hiện: <ol style="list-style-type: none"> Nhập vào số nguyên dương N ($N \leq 200$) từ bàn phím, tạo 2 ma trận kích thước $N \times N$ có giá trị ngẫu nhiên và lưu ra 2 file khác nhau. Đọc vào 2 ma trận từ file đã tạo, tính tích 2 ma trận trên và lưu kết quả ra file. Đo thời gian thực thi của hàm nhân ma trận sử dụng hàm clock() của thư viện time.h 	+ Mã nguồn C, với các file input, output theo yêu cầu. Code yêu cầu viết rõ ràng, có chú thích cụ thể.	7/4	12/4

	<p>c. Tìm hiểu về các thuật toán tối ưu nhân ma trận. Viết lại hàm nhân ma trận với một thuật toán tối ưu và đo thời gian thực thi.</p>	<p>+ Tài liệu mô tả thuật toán tối ưu đã tìm hiểu ở 3c.</p>		
4	<p>Bài tập 4: The tài liệu chuẩn 3GPP TS38.211, phần 7.4.1.1, chuỗi tín hiệu DMRS được tạo bằng công thức:</p> $r(n) = \frac{1}{\sqrt{2}}(1 - 2 \cdot c(2n)) + j \frac{1}{\sqrt{2}}(1 - 2 \cdot c(2n+1)).$ <p>Trong đó, chuỗi giả ngẫu nhiên $c(i)$ được định nghĩa theo mục 5.2.1, với c_{init} được cho bởi công thức:</p> $c_{init} = (2^{17}(N_{symb}^{slot} n_{s,f}^{\mu} + l + 1)(2N_{ID}^{n_{SCID}} + 1) + 2N_{ID}^{n_{SCID}} + n_{SCID}) \bmod 2^{31}$ <p>5.2.1 Pseudo-random sequence generation</p> <p>Generic pseudo-random sequences are defined by a length-31 Gold sequence. The output sequence $c(n)$ of length M_{PN}, where $n = 0, 1, \dots, M_{PN} - 1$, is defined by</p> $\begin{aligned} c(n) &= (x_1(n + N_C) + x_2(n + N_C)) \bmod 2 \\ x_1(n + 31) &= (x_1(n + 3) + x_1(n)) \bmod 2 \\ x_2(n + 31) &= (x_2(n + 3) + x_2(n + 2) + x_2(n + 1) + x_2(n)) \bmod 2 \end{aligned}$ <p>where $N_C = 1600$ and the first m-sequence $x_1(n)$ shall be initialized with $x_1(0) = 1, x_1(n) = 0, n = 1, 2, \dots, 30$. The initialization of the second m-sequence, $x_2(n)$, is denoted by $c_{init} = \sum_{i=0}^{30} x_2(i) \cdot 2^i$ with the value depending on the application of the sequence.</p> <p>Cho các giá trị: $M_{PN} = 1638, N_{symb}^{slot} = 14, n_{SCID} = 0, N_{ID}^{n_{SCID}} = N_{ID}^{CELL} = 1000$. Viết chương trình C thực hiện tạo chuỗi DMRS với các tham số đầu vào nhập từ bàn phím: slotIndex ($n_{s,f}^{\mu} \in [0, 19]$), symbolIndex ($l \in [0, 13]$). Dữ liệu chuỗi được tách riêng phần thực (I), phần ảo (Q) và đổi ra định dạng fixpoint Q2.14 (Q2.14_data =</p>	<p>Mã nguồn C, với các file input, output theo yêu cầu. Code yêu cầu viết rõ ràng, có chú thích cụ thể.</p>		

$\text{round}(\text{floating_input} * 2^{14})$. Lưu kết quả ra file có tên *dmrs_slot x _symbol y .txt* với định dạng: I1 Q1 I2 Q2.... Ví dụ định dạng file như hình sau:



The screenshot shows a Notepad++ window with the file *dmrs_slot1_sym3.txt* open. The file contains 17 lines of data, each with a line number followed by two pairs of integers representing I and Q values. The data is as follows:

Line	I1	Q1	I2	Q2
1	-11546	-11546		
2	11546	-11546		
3	-11546	11546		
4	-11546	-11546		
5	-11546	-11546		
6	-11546	11546		
7	-11546	11546		
8	11546	11546		
9	11546	11546		
10	-11546	11546		
11	-11546	-11546		
12	11546	11546		
13	11546	11546		
14	11546	-11546		
15	-11546	-11546		
16	-11546	-11546		
17	11546	-11546		

