

## Lab 10

ID1303: Introduction to Programming

1. Write a function that accepts a string and removes duplicates of characters.

Example runs:

Enter the string: bananas

Output string: bans

Enter the string: cataract

Output string: catr

2. Write a recursive function defined as `void permute(char *str, int n)` that prints `str` with all possible permutations of the first  $n$  characters. For example, if the string is `ABCDE` and  $n = 3$ , then the function should print `ABCDE,ACBDE,BACDE,BCADE,CABDE,CBADE` (not necessarily in this order).
3. Write a program that accepts a permutation of  $1, 2, \dots, n$  (stored in the array `str`) and finds the permutation that comes after `str` in the lexicographical ordering. For example, if the input sequence is `5, 3, 2, 4, 1` the next permutation is `5, 3, 4, 1, 2`. If the input sequence is the last one, eg: `5, 4, 3, 2, 1` you may output either the last one, eg: `5, 4, 3, 2, 1` or the first one, eg: `1, 2, 3, 4, 5`.
4. Create a structure called `Polynomial` to store the degree  $n$  and the coefficients  $a_0, a_1, \dots, a_n$  of a real polynomial  $P(x) = a_0 + a_1x + \dots + a_nx^n$ . Accept a polynomial  $P(x)$  and another polynomial  $Q(x)$  of degree two from the user and print the coefficients of the remainder when  $P(x)$  is divided by  $Q(x)$ .

Example run:

Enter the degree of the first polynomial: 3

Enter the value of a3: 1

Enter the value of a2: 1

Enter the value of a1: 1

Enter the value of a0: 1

Second polynomial, of degree 2

Enter the value of b2: 1

Enter the value of b1: -2

Enter the value of b0: -1

The remainder is  $8x+4$ .

5. The 15-puzzle is played as follows: There is a 4 by 4 matrix consisting of entries 1,2,...,9,A,B,C,D,E,F in 15 different positions and one empty position. The goal of the game is to start with an initial configuration of the grid and reach the configuration where the characters are in the right order by a sequence of moves, where each move consists of swapping the empty location with any of its neighboring values.

The matrix on the left is the desirable configuration. This can be reached from the configuration on the right by swapping the blank location value with B and then swapping the blank location value with C.

1	2	3	4
5	6	7	8
9	A	B	C
D	E	F	

1	2	3	4
5	6	7	8
9	A		B
D	E	F	C

Write a program that lets the user play the 15-puzzle. Display some initial configuration and accept a sequence of moves from the user and after each move, display the new matrix. The moves may be accepted as N,W,E or S, where N implies that the blank square has to be swapped with the square above it. W,E,S are self-explanatory. The program will stop if the user inputs the character X. Use zero (0) for the blank location.

6. Write a program to solve the  $n$ -queens problem, which consists of placing  $n$  queens on a  $n \times n$  chessboard (square matrix) so that no two attack each other (i.e no two are in the same row or column or diagonal).