

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Programming with C and C++ (CSC-101)

Assignment 04

Autumn Semester 2023-24

1. On your Smart Phone keypad, the alphabets are mapped to digits as follows: ABC(2), DEF(3), GHI(4), JKL(5), MNO(6), PQRS(7), TUV(8), WXYZ(9).
Write a C program called SmartPhoneKeyPad, which prompts the user for a String (case insensitive) and converts it to a sequence of Keypad digits. Use a nested-if (or switch case) in this exercise. Modify your program to use an array for table look-up later.
2. Write a C program that converts an English language to Piglatin. To do that, there are three rules: if the word starts with a vowel, add *way* to the end, Ex. **apple** = **appleway**. If the word has a vowel but doesn't start with it, then take the consonants in front of the first vowel and put them to the end of the word, and add *ay* to the end. Ex: **ball** = **allbay**, **strong** = **ongstray**, and if the word has no vowels, just add *ay* to the end. Ex. **pfift** = **pfiftay**.
Suppose the string is “**Proud to be an IITian**” then its corresponding **piglatin** string is “**Oudpray otay ebay anway IITianway**”.
(URL for piglatin translator:
<http://www.snowcrest.net/donnelly/piglatin.html>)
3. Write and test a function to print all the factorial numbers up to an input limit: (For example: if the input is 6; your program should print 1, 1, 2, 6, 24, 120, 720).
4. Write and test a function `int sum_digit (int n, int k)`. This function returns the sum of the k digit of the positive integer n. For e.g., if n is the positive integer 56789, then call `sum_digit(n,0)` would return 9, and the call `sum_digit(n,2)` will return 24.
5. Write and test a function to check that a given digit is present in a given number or not.
6. Consider the following recursive function and compute `f(11)`

```
int f(int i)
{
    if (i == 0) return 0;
    if ((i % 2) == 0)
        return f(i/2) + f(i/2);
    else return f(i-1) + 1;
}
```
7. Write and test a recursive function to convert a decimal into an octal number.
8. Write a function using pointers to find the Average and Median of *n* numbers. Validate your program using C.

9. In an array of n elements, write a C program to delete all the prime positions using pointers. For example: if the given array is 1.23, 45.6, 23.2, 45.4, 99.2, 35.4 then the resultant array should be 1.23, 45.4, 35.4.
10. Given an array of n elements. Insert a given number in the i^{th} position of the array where the starting position is zero. Use the concept of arrays and pointers.
11. Write a program that does the following:
 - Accept from screen the integer coefficients a , b , and c and the constant d of 3 linear equations in three variables, one line per equation.
 - Store the coefficients in a coefficient matrix which is declared as a 3×3 matrix and the constants (d) in a one-dimensional constant array of 3 elements.
 - Find the solution of the system of equations using a function known to you and display the equations accepted and the computed solution.
12. Write a C program to find all the palindromic substrings in a given string and print the start and end indexes of each of these substrings. A palindrome is a string that reads the same forward and backward (ignoring spaces, punctuation, and capitalization).

Input:

 - A string 'S' of length 'N' ($1 \leq N \leq 1000$), consisting of only alphabets (lowercase and uppercase).

Output:

 - For each palindromic substring found, print the substring, start index, and end index in the format: 'Substring: Start Index - End Index'.
 - If no palindromic substring is found, print "No Palindromic Substrings Found".

Example:

Input:

racecarlevel

Output:

racecar: 0 - 6

aceca: 1 - 5

cec: 2 - 4

level: 7 - 11

eve: 8 - 10
13. Write a program that implements a pattern recognition algorithm on a 2D array of integers. Given a 2D array (matrix) of integers, your program should find and print all 'cross patterns' within the matrix. A 'cross pattern' is defined as a center integer that is strictly greater than its immediate neighbors (above, below, left, and right). For example:

```
3 2 1
1 3 2
1 2 1
```

In this matrix, '3' at position '(1,1)' forms a cross pattern, because it's greater than its neighbors '2', '2', '1', and '2'.

Input:

- The first line contains two integers 'm' and 'n' representing the rows and columns of the matrix respectively ($3 \leq m, n \leq 50$).
- Following 'm' lines contain 'n' integers separated by spaces.

Output:

- If cross patterns are found, print their coordinates in the format '(row, column)'.
- If no pattern is found, print "No cross patterns found".

Example:

Input:

```
3 3
3 2 1
1 3 2
1 2 1
```

Output:

(1,1)

14. You are given an array of strings. Write a program that transforms each string in the array according to the following rules:
1. If the length of the string is odd and greater than 4, reverse the string.
 2. If the length of the string is even and greater than or equal to 4, swap the first half and the second half of the string.
 3. If the string length is 1 or 2 or 3, convert all characters to uppercase.

After the transformations, print the final array of strings.

Input:

- The first line contains an integer 'N' ($1 \leq N \leq 100$), the number of strings in the array.
- The next 'N' lines contain the strings, each string is non-empty and has a length of at most 100 characters.

Output:

Print the 'N' transformed strings, each on a new line.

Example:

Input:

5

hello

world

cat

ab

APPLEE

Output:

olleh

dlrow

CAT

AB

EAPPL

Explanation:

- “hello” has an odd length of 5, so it is reversed to become “olleh”.
- “world” also has an odd length, so it is reversed to become “dlrow”.
- “cat” has a length of 3, so it is converted to uppercase to become “CAT”.
- “ab” has a length of 2, so it is converted to uppercase to become “AB”.
- “APPLEE” has an even length of 6 and is greater than or equal to 4, so its first half “APP” and second half “LEE” are swapped to become “LEEAPP”.

15. You are given an $N \times M$ matrix consisting of 0’s and 1’s. Your task is to implement a program to find and print a path from the top-left corner to the bottom-right corner. You can only move right or down, and you can only move through cells containing 1. If there’s no such path, your program should indicate so.

Input:

- The first line contains two integers N and M ($1 \leq N, M \leq 100$), the size of the matrix.
- The next N lines contain M integers (0 or 1), representing the cells of the matrix.

Output:

- If there is a path, print the path as a sequence of directions (“R” for right, “D” for down).
- If there’s no path, print “No path found”.

Example:

Input:

```

3 3
1 0 1
1 1 1
0 1 1

```

Output:

DRRDD

Explanation:

In the given matrix:

```

1 0 1
1 1 1
0 1 1

```

The path from the top-left to the bottom-right through the cells with 1's is Down, Right, Right, Down, Down.

16. Given two matrices 'A' and 'B', your task is to write a program that:

1. Calculates the dot product of the matrices, ' $C = A \cdot B$ '.
2. Transposes matrix 'C' to get matrix 'D'.
3. Replaces every element ' $D[i][j]$ ' in matrix 'D' with the absolute difference between ' $D[i][j]$ ' and the average of its neighboring elements. For the purpose of this problem, "neighboring elements" are the elements horizontally and vertically adjacent to the given element. Apply least integer function to each computation in the end.

Input:

The first line contains a single integer 'N' ($1 \leq N \leq 100$), denoting the order of the square matrices 'A' and 'B' (i.e., both matrices are of size 'N x N').

Next 'N' lines contain the elements of matrix 'A', followed by 'N' lines for matrix 'B'.

Output:

Print the transformed matrix 'D' of size 'N x N'.

Example:

Input:

```

2
1 2
3 4
1 0
0 1

```

Output:

2 1
1 2

Explanation:

Matrix 'C = A . B':

1 2
3 4

Transpose to get matrix 'D':

1 3
2 4

For the element 'D[1][1] = 1', the neighboring elements are 3 and 2. Their average is '(3 + 2) / 2 = 2.5'. The absolute difference between 1 and 2.5 is 1.5, which we'll raise to 2. Hence, the new value of 'D[1][1]' becomes 2.

Applying similar operations for other elements will give us:

2 1
1 2

[Submission Link](#)