

CS19001: Programming and Data Structures Laboratory

Soumyajit Dey, Aritra Hazra
CSE, IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2018/CS19101_PDS-Lab_Autumn2018.html

22-Oct-2018

Programming Assignments

Complete and submit during lab

Assignment 1: [Mind-Your-Words]

- Initialize a 2-d character array comprising 20 rows and 30 columns. Read up 20 words from the keyboard and store them.
- Lexicographically sort the 2-d array and print all the words in that order.
- Display the number of words that are of length between 1-2 letters, 3-5 letters, larger than 5 letters.
- Find and display the distribution (percentage) of the letters a to z in proper format by considering all the words together.
- Find all duplicate words, remove them, bring the succeeding words forward and display the updated sorted list of words.

Assignment 2: [Target Standard Deviation]

Recall that (for five data points a, b, c, d, e)

- Arithmetic mean $AM = \frac{(a+b+c+d+e)}{5}$

- Standard deviation $SD = \sqrt{\frac{a^2+b^2+c^2+d^2+e^2}{5} - (AM)^2}$

Implement a C function

```
int standard_dev(int a[ ], int i, int k, double m, int *n)
```

which takes as argument a pointer to an integer sequence of length i . The function computes how many k length subsequences exist in the overall i length sequence for which the standard deviation (SD) is $\geq m$ and returns the value. The function writes the starting index of the k length subsequence whose SD is maximum to the memory location pointed to by n .

Assignment 2: [Target Standard Deviation]

Write a main() which

- declares an integer pointer, asks the user for an input size (i), dynamically allocates memory to the pointer, takes as input i integers and stores in the allocated memory.
- requests the user to provide values for k (subsequence size) and m (SD value).
- reports back the no. of k length subsequences for which the standard deviation (SD) is $\geq m$ and the starting index of the k length subsequence whose SD is maximum.

Assignment 3: [String Orientation]

Implement the following C functions

- `int string_order(char *s)` which takes as input a character pointer 's' and returns '1' if 's' is pointing to a string which is alphabetically ordered. The function assumes that the string contains all characters in lower case.
- `int sub_string_order(char *s, int k)` which takes as input a character pointer 's' and an integer k , finds out how many alphabetically ordered substrings of size k exist in the string pointed to by 's' and returns the value.

Assignment 3: [String Orientation]

Write a main() which

- declares a character pointer, asks the user for a string size (n), dynamically allocates memory to the pointer and takes a lowercase string as input (properly terminated by a null character).
- requests the user to specify an integer $k \leq n$.
- reports back the no. of alphabetically ordered substrings of size k in the user input string.

Thank You