

DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

Assignment Set: 1

Problem No: 5

Problem Statement:

Write a program to generate 1,00,000 random strings of capital letters of length 10 each, without repetitions and store them in a file in character mode one string per line.

Solution Approach:

The system time changes every second, and this is the property that is used in the following solution where `srand()` and `rand()` together with a `time_t` variable is called. A character array containing the 26 alphabets is used to find different permutations and hence generate non-duplicate words.

Inside a loop the `rand()%100000` generates a random number less than or equal to 100000. For each iteration of the outer loop an inner loop is ran in which a letter is chosen at random from the character array and appended at the end of the string i.e to be written in the file.

A binary array is made which stores 1 if that index is already present in the file, 0 otherwise. A check is made if the number is repeated in which case, the loop variable is repeated, written to the file otherwise.

A binary file (also attached below) is used to store the random numbers.

DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

Structured Pseudocode:

FILE *PTR = FOPEN(FILE_NAME, MODE)

TIME_T VAR

CHAR ARRAY[26]={‘A’, ‘B’, ‘C’,....., ‘Y’, ‘Z’}

ARR[10000] = {0}

SRAND (VAR)

FOR i=1 TO 100000 DO:

 INT K=RAND()%100000

 FOR j=0 TO 9 DO:

 RESULT[j]=ARRAY[RAND()%26]

 IF (K NOT IN FILE)

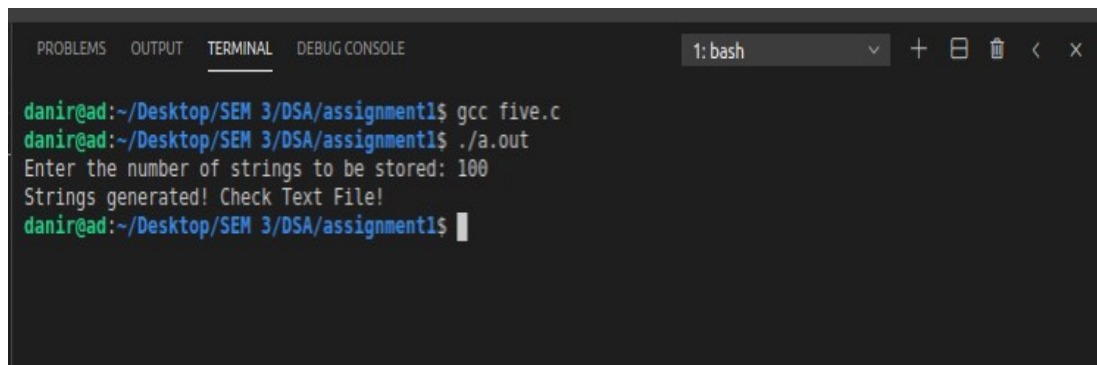
 FPRINTF(PTR,K);

 ELSE

 i = i-1

 ARR[i]=1

Results:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ gcc five.c
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ ./a.out
Enter the number of strings to be stored: 100
Strings generated! Check Text File!
danir@ad:~/Desktop/SEM 3/DSA/assignment1$
```

DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

Discussions:

The solution approach uses $O(n)$ time complexity. The space complexity is $O(2*n) - O(n)$.

Source Code:

FILE NAME:

Code – “five.c”

Binary File – “assign4.txt”

(can be found in the following link: [https://drive.google.com/drive/folders/1-](https://drive.google.com/drive/folders/1-nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing)

[nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing](https://drive.google.com/drive/folders/1-nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing))