

DSA REPORT

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

Assignment Set: 1

Problem No: 8

Problem Statement:

Convert your Name and Surname into large integers by juxtaposing integer ASCII codes for alphabet. Print the corresponding converted integer. Cut the large integers into two halves and add the two halves. Compute the remainder after dividing the by the prime numbers P in problem 7.

Solution Approach:

The user entered name is stored in an integer array to get the ascii values of individual characters. This integer array is modified to store 2-digit ascii codes as two separate integers to avoid overflow in integer value. The ascii array is then printed. Next this ascii array is separated into two halves and stored in two integer arrays which are printed.

These two large integer halves are now added (arithmatically), and the result is stored in another array. A carry variable is used to store the carry (if encountered) while adding the digits and is finally added to the last element of the addition array.

Next first eight 4-digit prime numbers are generated and stored in a prime array. The added large integer is then divided by each of the eight prime numbers and the remainder is stored and printed.

Note: Functions for checking prime and calculating remainder using modular arithmetic are defined separately.

DSA REPORT

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

Structured Pseudocode:

FUNCTION PRIMR(INT N):

FOR i=2 TO N-1 DO:

IF (N % i == 0)

RETURN 0

i = i*i

RETURN 1

FUNCTION REMINDER(INT A, INT B, INT M):

A = A%M

RES = 1

WHILE B>0 DO:

IF(B IS ODD):

RES = RES*A %M

A = A*A %M

B = B/2

RETURN RES

MAIN():

INPUT NAME

INT ARRAY1[]

ARRAY[i] = NAME[i] FOR ALL CHARACTERS OF NAME

ASCII[2*i] = ARRAY[i]/10 ASCII[2*i +1] = ARRAY[i]%10

PRINT ASCII[j] FOR ALL i=0 TO i < LENGTH(ASCII)

STORE FIRST HALF OF ASCII IN ARRAY 'A'

STORE SECOND HALF OF ASCII IN ARRAY 'B'

DSA REPORT

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

PRINT A,B

FINAL[i] = (A[i] + B[i] + CARRY)%10 FOR ALL i=0 TO LENGTH(A)

CARRY = (A[LENGTH(A)-1] + B[LENGTH(A)-1] + CARRY)/10

PRINT FINAL

FOR i=1001 DO:

IF (COUNT<8)

IF (PRIME[i] == 1)

ARRAY[COUNT] = i

COUNT = COUNT +1

i = i+1

FOR i=0 TO 8 DO:

REM=0

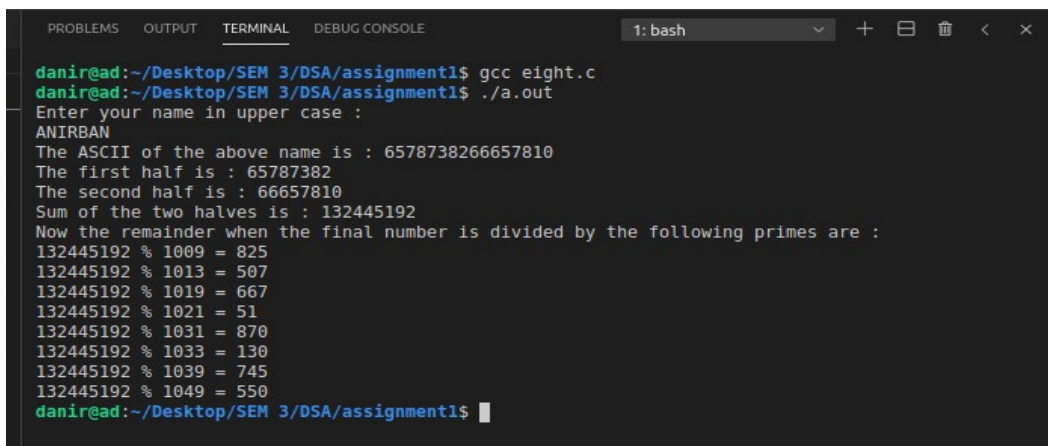
FOR j=0 TO LENGTH(FINAL) DO:

VAR = (FINAL[j] * REMINDER(10, j, PRIMEi))) % PRIME[i]

REM = (REM+VAR) % PRIME[i]

PRINT PRIME[i], REM

Results:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ gcc eight.c
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ ./a.out
Enter your name in upper case :
ANIRBAN
The ASCII of the above name is : 6578738266657810
The first half is : 65787382
The second half is : 66657810
Sum of the two halves is : 132445192
Now the remainder when the final number is divided by the following primes are :
132445192 % 1009 = 825
132445192 % 1013 = 507
132445192 % 1019 = 667
132445192 % 1021 = 51
132445192 % 1031 = 870
132445192 % 1033 = 130
132445192 % 1039 = 745
132445192 % 1049 = 550
danir@ad:~/Desktop/SEM 3/DSA/assignment1$
```

DSA REPORT

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

Discussions:

Most of the functions use $O(n)$ time complexity, the exception being the modular exponentiation which uses $O(\log n)$. Therefore, in the worst case the complexity is $O(n)$. The space complexity is $O(n)$.

Source Code:

FILE NAME:

Code – “eight.c”

(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))