Indian Institute of Technology, Kharagpur

CS19001 Programming and Data Structures Laboratory, Autumn 2019

# Assignment for Week 7 (October 14, 2019)

**Total Marks:** 40 **Submission Deadline:** 17:45

## INSTRUCTIONS

1. Submit a separate C file for each of the problems. The solution for problem $i$ should be named **[rollno]-prob*i*.c** where '[rollno]' is your roll number.
2. You may consult your notes, books or manual pages.

## PROBLEMS

1. Write a program to maintain a table of variable-value pairs. The table is to be maintained as an array.

   You will need to define a structure for an entry in the dictionary, and another structure for storing the dictionary.

```
typedef struct entry {
        char var[40];
        char * val ;
}  dictnode;

typedef  struct {
        dictnode *dictionary[100] ;
        int numentry ;
} DICT;
```

   You must define the following functions:

```
DICT * createdict ( ) ;
             // This function allocates space to create a new dictionary and returns
             // a pointer to it.
dictnode makeentry ( char  var [ ], char val[ ]) ;
             // This function creates a new dictionary entry
void addlentry (dictnode newentry,  DICT * pdictionary ) ;
             // This function adds  a new entry to the dictionary.
             // If an entry with the same var name exists, the val field is modified.
             // Otherwise a new entry is created
void printentry (DICT * pdictionary, char var[]) ;
void delentry  (DICT * pdictionary, char var[]) ;
             // Delete the entry from the dictionary
void printdict (DICT * pdictionary) ;
```

   Write a menu driven C program that calls the above functions. Your program must accept the following commands. Note that the pointy brackets, < and > indicate user defined input and are not typed in the input.

   (a) add $< var > < val >$ : $< var >$ can be any string. $< val >$ can be any integer. Create an entry with the corresponding values and add it to the dictionary.

(b) find $< var >$ : If there is an entry with a var field that contains the given string, print the entire entry. If not, print ”not found”.

(c) del $< var >$ : Delete the entry of the dictionary corresponding to the var.

(d) print : Print the dictionary as it exists so far.

(e) quit

**Examples**

```
add name rohit
add surname prasad
add name ritesh
add book1 gonewiththewind
find name
        name ritesh
find book2
        book2 not found
del name
add book2 homosapiens
add name sumana
print
        name sumana
        surname prasad
        book1 gonewiththewind
        book2 homosapiens
quit
        Bye.
```

Marks: 15

2. Write a function `reverse_int(int a, int* len)` that returns an integer that contains `a`’s bits in reverse order and stores the bit length of `a` in `*len`. Use this function to print the binary representation of a given non-negative integer.

**Examples**

(a)
```
-87
Input should be non-negative.
```

(b)
```
14
The binary representation of 14 is 1110
```

(c)
```
32456
The binary representation of 32456 is 111111011001000
```

Marks: 15

3. The sum $1 + \frac{1}{2} + \frac{1}{3} + \cdots$ does not converge and becomes infinitely large. But if it is calculated with finite precision by a computer, the sum actually exists since the terms eventually get so small that when represented in finite precision they are essentially 0. Suppose we calculate the sum by rounding to 1 decimal place. Then we have $1+0.5+0.3+0.3+0.2+0.2+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1 = 3.9$. More precisely, let $r_n(x)$ be the number $x$ rounded to $n$ decimal places; we define

$$r_n(x) = \lfloor 10^n x + 0.5 \rfloor / 10^n.$$

Then we wish to find

$$S_n = r_n(1) + r_n\left(\frac{1}{2}\right) + r_n\left(\frac{1}{3}\right) + \cdots$$

We know that $S_1 = 3.9$ and your program should calculate and print $S_n$ for $n = 2, 3, 4, 5$.

**Expected output**

```
S2 = 6.16
S3 = 8.449
S4 = 10.7509
S5 = 13.05363
```

Marks: 10