

Data Structure Assignment [5]

General information

- Deadline: **2020/11/19 12:00 noon time.**
- Submit your programming assignment to the Moodle system.
- Submitted file format: student-ID Name.zip (e.g. F12345678_王曉明.zip.) Please name the filename of your submitted compressed file after your **student ID number. Otherwise, 20 points will be deducted.**
- Your submitted files have to be organized in a directory-structured manner as follows. Otherwise, **5 points will be deducted.**
 - | -- F12345678_王曉明
 - | -- F12345678_王曉明.pdf
 - | -- code
 - | -- xxxxx.c
 - | -- xxxxx.c
- Your submitted file must contain **Source Code & Readme file** (Program description in .pdf format)
- **Late submission will not be accepted.**
- There is a “zero tolerance” for plagiarism. You will receive a score of zero if you get caught plagiarizing.

Course Provisions

1. Program execution environment : Windows (Git Bash)
Command: `./hw.exe < input.txt > output.txt`
2. Programming language: C (standard: C11)
(**Languages other than C are not accepted**)
3. Submitted programming homework must include **source code** in .c data type, and **readme document** in .pdf data type. You are required to address the **(1) result screenshot, (2) program structure, (3) program functions** in readme file. Do not just write the pseudo code or even just copy and paste your code!
4. Homework grading is divided into two parts: 80% for the code and 20% for the readme file. The remaining grading standards are determined by the TAs.

TA office hours of the course:

Wed. 11:30 – 13:00

Lab location: CSIE Bldg. Room 65302

If you have any question, please make an appointment in advance.

You can also mail us about your questions.

TA e-mail: ta_@dblab.csie.ncku.edu.tw

Programming homework

Double Hashing Mechanism

In this assignment, you are required to implement a double hash mechanism. Its hash function is $(\text{hash1}(\text{key}) + i * \text{hash2}(\text{key})) \% \text{TABLE_SIZE}$, in which $\text{hash1}()$ and $\text{hash2}()$ are two individual hash functions defined as follows.

- $\text{hash1}(\text{key}) = \text{key} \% \text{TABLE_SIZE}$
- $\text{hash2}(\text{key}) = \text{PRIME} - (\text{key} \% \text{PRIME})$

TABLE_SIZE is the size of hash table which in this homework is 13. In others word, each row of the hash table (i.e., each bucket) has a room for only one key. PRIME is a prime number smaller than TABLE_SIZE which in this homework is 7.

The INPUT file will contain one and exactly only one row of keys separated with a space, as shown in the following INPUT. Hash these keys into the hash table according to the order of the keys given in the INPUT file. In the beginning, i is set to 0. When a collision occurs during hashing a key, recalculate its hash value by $i = i + 1$. If collision still occurs, then recalculate its hash value again by setting $i = i + 1$. Repeat this process until no collision occurs (i.e., an empty bucket is found), and then put the key in that bucket. Then, we continue to hash the next key. When hashing the next key, i is reset to 0 and the same hash process repeats for this new key. This hashing process shall repeat until all keys in the INPUT file are hashed into the hash table.

NOTE:

- After finishing the entire hashing process, print out the keys in the hash table. Your output format must be exactly the same as the hash table shown in the following “Output”, in which the “index->key” must be printed in each line. Neither extra empty space nor extra empty line is accepted. They will be considered incorrect answer if they appear.
- Just use `scanf()` to get the input content and `printf()` to print out the results.

In terms of the format of the INPUT & OUTPUT, please refer to the input_windows.txt & output_windows.txt in HW5.zip for details.

Execution:

Your homework must use the following command to execute in the **Windows Git Bash terminal**, and your output file format must be the same as that in the given output file. In addition to the given input data files, there will be another two hidden input data files used to grade your homework.

Step 1. `gcc -std=c11 ./*.c -o hw5`

Step 2. `./hw5.exe < input0_windows.txt > ans_output0_windows.txt`

Step 3. `diff ./output0_windows.txt ./ans_output0_windows.txt`

Input

```
24 10 31 56 45 85 64 8 77 37 2 98 70
```

Output

```
0->77
1->98
2->2
3->37
4->56
5->31
6->45
7->85
8->8
9->70
10->10
11->24
12->64
```