PHW5 – 1

#define \_CRT\_SECURE\_NO\_WARNINGS

#define size 33

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct HashTable {

int value;

};

struct HashTable HashTables[33];

void insert(int value)

{

int key = value % size;

if (HashTables[key].value == -1)

{

HashTables[key].value = value;

printf("%d inserted at arr[%d]\n", value, key);

}

else

{

printf("Collision : arr[%d] has element %d already!\n", key, HashTables[key].value);

printf("Unable to insert %d\n", value);

}

}

int main() {

FILE\* ptr;

// Opening file in reading mode

ptr = fopen("keyinput.txt", "r");

if (NULL == ptr) {

printf("file can't be opened \n");

}

int i=0;

const int max = 1024;

char line[max];

char\* pLine;

//FILE\* in = fopen("in.txt", "r");

int count = 0;

while (!feof(ptr)) {

pLine = fgets(line, max, ptr);

//printf("%s", pLine);

int sum = 0, i, len;

len = strlen(pLine);

for (i = 0; i < len; i++)

{

sum = sum + pLine[i];

}

//printf("Sum of all characters : %d", sum);

if(i<3) insert(sum);

i++;

}

fclose(ptr);

return 0;

}

PHW5 – 2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct T1 {

int value;

};

struct T1 Table1[15][3];

struct T2 {

int value;

};

struct T2 Table1[10][3];

int main() {

return 0;

}