­CSCI 3232 Systems Software Assignment 3

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Feb 6, Thursday, 2020. Note: Make all your source codes compilable and runnable under Ubuntu. Do NOT put your codes in Word or PDF documents. Make them into separate text files with appropriate file extensions (.h, .c, etc) as you would compile them.**

1. (40 pts) Following the example C program in Folio consisting of three files (a.c, a.h, main.c), write a C program that consists of three files ***mysquare.h***, ***mysquare.c*** and ***myMain.c.*** Below is the ***mysquare.h*** prototype

#include <stdlib.h>

#include <stdio.h>

void generateNums(int \*myarr, int len);

void squareNums(int \*myarr, int len);

void printNums(int \*myarr, int len);

* **mysquare.h must contain only the function declarations (prototypes) listed above**
* **generateNums** function should generate **len** random integers in the range 0~99 inclusive (use *rand* function in <stdlib.h>) in the supplied array **myarr.** Its definition should be in mysquare.c.
* **squareNums** function should square the **len** integers stored at **myarr**. Its definition should be in mysquare.c.
* **printNums** function should print the **len** integers stored at **myarr** on one line with suitable separators between them**.** Its definition should be in mysquare.c.
* Note that the caller of the above three functions should pass valid pointers myarr to them.

1. In myMain.c you need to write your *main* function that calls the three functions in mysquare.h to generate N random integers in the range 0~99 inclusive and then print out the generated random numbers on one line and the squared N numbers on another line. Your program should get N(which represents the number of random integers you will generate) from the first command line argument (i.e. argv[1]) of this C program (you can use the *atoi* function; see example code LeapYear\_cmdline.c). You need to decide where to store the N generated integers and where to store the N squared integers. Submit all the three code files. (37 pts)
2. Submit a makefile to compile your program. Be aware of the possible file name issue regarding Folio complaint. See the last slide of 3\_Pointers\_Functions.pptx for details. Make sure you have tested that your makefile works. (3 pts)

2. (40 pts) You need to understand array, pointer and pointer arithmetic to complete this exercise. And you should be able to figure out the answers without actually compiling and running the following C program.

#include <stdio.h>

int main(int argc, char \*argv[]) {

char a, \*pc, c[9];

int i, \*pk, k[9];

a='z';

pc=&(c[8]);

pk=&(k[0]);

for (i=0; i<9; i++) {

\*pc=a-(char)i;

pc--;

\*pk=(int)a-i;

pk++;

}

return 0;

}//end of main

1. Write out the memory map for the above C program in the following table. For array variables, list the address range for the entire array. Assume the memory address starts from 100, that is, the address for **a** is 100. Suppose sizeof(**char**)=1, sizeof(**int**)=4 and the size of a memory address (a pointer) is 8 bytes. You may assume there is no gap in memory between the variables. (18 pts)

|  |  |  |
| --- | --- | --- |
| Variable | Start address | End address |
| a | 100 | 100 |
| pc | 101 | 37 |
| c | 109 | 109 |
| i | 118 | 118 |
| pk | 122 | 186 |
| k | 130 | 130 |

1. Show values of the variables in the following table at the end of execution of the *for* loop before the main function returns. (For the two array variables c and k, list the contents of all array elements.) (18 pts)

|  |  |
| --- | --- |
| Variable | Value |
| a | z |
| pc | \*(c-1) |
| c | r | s | t | u | v | w | x | y | z | |
| i | 9 |
| pk | \*(k+9) |
| k | 122 | 121 | 120 | 119 | 118 | 117 | 116 | 115 | 114 | |

1. True or false? pc and pk have the same value just before return. (2 pts)

false.

1. True or false? Not taking order of the elements into consideration, the two array c and k contain the same integral values just before return. (2 pts)

false.

3. (8 pts) Suppose we have the following piece of C code. Which of the following statements is illegal?

struct Books //define the struct Books

{

char title[50];

char author[50];

char subject[100];

int book\_id;

} Book1;

struct Books book, \*book2;

1. Book1=book; b. book.book\_id=Book1.book\_id; c. book.title=Book1.title;

d. book2=&Book1; e. book2->author f. (\*book2).subject

c. book.title=Book1.title

4. (12 pts) This exercise further tests your knowledge of pointer arithmetic. Suppose we have three variables declared as

char \*pc;

int \*pi;

struct point {double x; double y;};

struct point \*p1;

struct point \*p2[10];

Assume sizeof(char)=1, sizeof(int)=4, sizeof(double)=8. The values of pc, pi, p1 and p2 are 240, 258, 410 and 480 respectively.

What are the values of pc+1, pi+2, p1+4 (12 pts) ?

241, 276, 474

What is the value of p2+3 (4 pts extra credit)?

528

checklist of five files to be submitted: one solution file, mysquare.h, mysquare.c, myMain.c, makefile