CSCI 3232 Systems Software Assignment 3

Upload all your files to the dropbox in Folio before the deadline --- **11:30PM Feb 12, Tuesday, 2019. 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 points) Learn the example C program in Folio consisting of three files (a.c, a.h, main.c) and complete this exercise. You need to understand function and pointer and know the selection sorting algorithm. Write a C program that consists of three files ***mysort.h***, ***mysort.c*** and ***myMain.c.*** Below is the ***mysort.h*** prototype

#include <stdlib.h>

#include <stdio.h>

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

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

* **mysort.h must contain only the function declarations (prototypes) listed above**
* **generateNums** function should generate **len** random integers in the range 0~100 inclusive (use *rand* function in <stdlib.h>) in the supplied array **myarr.** Its definition should be in mysort.c.
* **sortNums** function should sort the **len** integers at **myarr** into ascending order using selection sort. Its definition should be in mysort.c. Note that the caller of sortNums and generateNums should pass valid pointers myarr to them.

1. In myMain.c you need to write your *main* function that calls the two functions in mysort.h to generate N random integers in the range 0~100 inclusive and then print out the generated unsorted random numbers in one line with suitable separators between them and the sorted N numbers in another line. Your program should get N(which represents the number of random integers you will generate and sort) 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). Submit all three source files. (37 points)
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 points)

2. (40 points) 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 program.

#include <stdio.h>

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

{

char a, \*pc, c[10];

int i, \*pk, k[10];

a='p';

pc=&(c[9]);

pk=&(k[0]);

for (i=0; i<10; i++)

{

\*pc=a-(char)i;

pc--;

\*pk=i+3;

pk++;

}

return 0;

}//end of main

1. Write out the memory map for the above C code 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 points)

|  |  |  |
| --- | --- | --- |
| Variable | Start address | End address |
| a | 100 | 100 |
| pc | 101 | 108 |
| c | 109 | 118 |
| i | 119 | 122 |
| pk | 123 | 130 |
| k | 131 | 170 |

1. Additionally, 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 points)

|  |  |
| --- | --- |
| Variable | Value |
| a | p |
| pc | c[0] |
| c | From c[0] to c[9]: g, h, i, j, k, l, m, n, o, p |
| i | 9 |
| pk | k[9] |
| k | From k[0] to k[9]: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 |

1. How many bytes in total does the code allocate for the variables? (4 points) 71 bytes

3. (8 points) You need to understand how to define and use a struct for this exercise. You should be able to figure out the answer without actually compiling or running the program. Which is true about the following codes? If you choose a, you need to specify where the syntax error(s) occur; if you choose b, you need to specify what error occurs when you run it; if you choose c, you need to specify the outputs of the program.

#include <stdio.h>

#include <string.h>

struct Books //define the struct Books

{

char title[50];

char author[50];

char subject[100];

int book\_id;

} Book1;

struct Books book;

void DealBooks(struct Books b1, struct Books \*p2)

{

b1.book\_id++; p2->book\_id++;

printf("%d\n",b1.book\_id);

printf("%d\n",p2->book\_id);

printf("%s\n",b1.title);

printf("%s\n",p2.title);

}

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

{

strcpy(book.title,"C Programming");

strcpy(book.author,"Nuha Ali");

strcpy(book.subject,"C Programming Tutorial");

book.book\_id=100;

strcpy(Book1.title,"Telecom Billing");

strcpy(Book1.author,"Zara Ali");

strcpy(Book1.subject,"Telecom Billing Tutorial");

Book1.book\_id=700;

DealBooks(book,&Book1);

printf("%d\n",book.book\_id);

printf("%d\n",Book1.book\_id);

return 0;

}//end of main

1. Compile error b. Compiles OK, runtime error c. Compiles and runs OK

4. (12 points; additional 4 points of extra credit for p2+3) 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, 310 and 480 respectively. What are the values of pc+1, pi+2, p1+4 (12 points) and p2+3 (4 points extra credit)? Pc+1= 241, pi+2= 266, p1+4=374, p2+3=504

2a)

|  |  |  |
| --- | --- | --- |
| Variable | Start address | End address |
| a | 100 | 100 |
| pc | 101 | 108 |
| c | 109 | 118 |
| i | 119 | 122 |
| pk | 123 | 130 |
| k | 131 | 170 |

2b)

|  |  |
| --- | --- |
| Variable | Value |
| a | p |
| pc | c[0] |
| c | From c[0] to c[9]: g, h, i, j, k, l, m, n, o, p |
| i | 9 |
| pk | k[9] |
| k | From k[0] to k[9]: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 |

2c) 71 bytes

3) The answer is a. p2.title is an invalid. You should either dereference or use the syntax p2->title

4) pc+1= 241, pi+2= 266, p1+4=374, p2+3=504