CSCI 152 Programming Fundamentals II Spring 2018

Dr. Creider Final Exam 5/07/2018

110 points

Your Name: Collin Quigley

**Type your name above**

Type your answers to the questions on the exam in this document. Save this file with the file name of ‘LastNameFirstName.doc’ and upload the file to the Final Exam drop box. The last question on this exam will require you to write 3 complete functions. You are permitted to use your notes on this exam. As a result, grading will be stricter. You can also use the compiler on the coding section of the exam.

Write the answers to the following questions using the variable definitions that follow.

struct product // structure definition – requires 10 bytes of memory

{ short length, width, height, ColorCode, weight}

product \*OilFilters, shoes, PaintScraper, battery, \*cookies;

long long LLvalue1, \*ptrLLV1;

long Lvalue1, \*ptrLV1, Lvalue2, \*ptrLV2;

short Svalue1, \*ptrSV1, Ssize=1000;

float Fvalue1, \*ptrFV1;

1. What is the purpose of the ‘\*’ operator in the declaration statements above? (2 points)

To create a pointer of that data type.

2. What is the purpose of the ‘\*’ operator in the following statement? (2 points)

cout<<\*ptrSV1++<<endl;

To display the value the pointer ptrSV1 is pointing to.

3. For each of the following pointers enter the size of memory in bytes required for the pointer. (10 points)

For a 32 bit operating system For a 64 bit operating system

32 bit compiler 64 bit compiler

OilFilters size= 4 size= 8

ptrLLV1 size= 4 size= 8

ptrLV1 size= 4 size= 8

ptrSV1 size= 4 size= 8

ptrFV1 size= 4 size= 8

4a. Write one statement to dynamically allocate an array of *Ssize* elements and store the address in the pointer *OilFilters*. (2 points)

OilFilters = new(nothrow)product[SSize];

4b. Write one statement to dynamically return to the memory manager the memory allocated in part a. (2 points)

delete [] OilFilters;

5. Write one statement to dynamically allocate an array of *Ssize* elements and store the address in the pointer *ptrLLV1.* (2 points)

ptrLLV1 = new(nothrow)long long[SSize];

6. What is the purpose of the ‘++’ operator in the following statement? (2 points)

cout<<\*ptrSV1++<<endl;

It is used to add 1 to value ptrSV1 points to after it is displayed.

7. Write an output statement to display the value stored in the *width* field for the variable *battery*. (2 points)

cout<< battery.width;

8. Write an output statement to display the value stored in the *ColorCode* field for element 10 of array *OilFilters*.

(2 points)

cout<< OilFilters[10].ColorCode;

9. Write an output statement to display the value stored in the *weight* field for the variable *cookies*. (2 points)

cout<< cookies->weight;

10. Write an output statement to display the value stored in the *length* field for the variable *shoes*. (2 points)

cout<< shoes.length;

11. For each of the following expressions below, select the appropriate result of the expression – A, I, or S

A (address value), I (integer value), S (syntax error);

NOTE: pointer is a pointer variable (type does not matter), offset is an integer variable (type does not matter) (18 points)

pointer + 25 Ans: A

pointer – 5 Ans: A

pointer + offset Ans: A

pointer \* offset Ans: S

pointer – pointer Ans: I

pointer + pointer Ans: S

pointer \* pointer Ans: S

pointer / 5 Ans: S

pointer – offset Ans: A

12. For each of the following expressions below, select the appropriate action – **P**, **V**, **B**, **N** or **S** (12 points)

P (pointer is incremented), V (value at address the pointer contains is incremented), B (both pointer and the value at the address the pointer contains are incremented), N (neither is incremented), S (syntax error).

(\*ptr)++ Ans: V

\*++ptr Ans: S

++(\*ptr) Ans: V

\*(++ptr)++ Ans: B

++\*ptr Ans: S

\*ptr++ Ans: V

13. Given the following declarations. (20 points)

struct product // structure definition – requires 10 bytes of memory

{ short length, width, height, ColorCode, weight}

product \*OilFilters, shoes, PaintScraper, battery, \*cookies;

short size, big, small, m;

For each of the following statements enter the correct response to indicate that the statement is valid or not valid syntax. Y it is valid, or N it is not valid.

shoes = OilFilters Ans: Y

\*cookies.ColorCode Ans: Y

\*OilFilters->length Ans: N

(\*(OilFilters+m)).height Ans: Y

battery.height Ans: Y

battery->ColorCode Ans: N

battery.\*height Ans: N

shoes = \* PaintScraper Ans: N

cookies[M].weight Ans: Y

OilFilters.length Ans: Y

\*cookies = shoes Ans: Y

shoes = cookies Ans: N

OilFilters = cookies Ans: Y

\*OilFilters = \*cookies Ans: Y

shoes = battery.width Ans: N

\*OilFilters = cookies[8] Ans: Y

(OilFilters+m)->height Ans: Y

(\*(OilFilters+m))->height Ans: N

if(size < shoes.weight) Ans: Y

if(battery> shoes) Ans: N

Three Function Definitions Required (50 points)

Given the code/program below, write three functions for this program. If you wish, you can copy this code to a compiler and test your code during the exam. Put your code for the functions immediately after the description of requirements for each function. **To get full credit all function must use pointers to access an array, and pointers must be used to terminate loops** (**NO integer controlled loops, no subscripts permitted, and no integer counters**) If you write the correct algorithm using subscripts you will receive no more than 1/2 partial credit. Arrays can be searched in any order.

a) Write a void function, input\_data, to dynamically allocate a short array of size elements and store the input data entered from the keyboard into the array. A pointer in which to store the base address of the array allocated is passed to the function as well as the size of the array to be allocated which was assigned a value in main. This function will also dynamically allocate the numbers array to be half the size of the data array. A separate pointer has been passed to the function for this array. Be sure that the addresses of the memory allocated in this function are available in main. (10 points)

void input\_data(short \*&data, short size, short \*&numbers)

{

short \*dataPtr, \*dataEnd;

data = new(nothrow)short[size];

if (!data) { cout<< "Memory allocation error, program will terminate\n"; system("pause"); exit(0); }

numbers = new(nothrow)short[size/2];

if (!numbers) { cout<< "Memory allocation error, program will terminate\n"; system("pause"); exit(0); }

dataPtr = data;

dataEnd = data + size;

while (dataPtr < dataEnd)

{

cin>> \*dataPtr;

dataPtr++;

}

}

b) Write a value returning function, *FindFourthDuplicate*, to find the **address of the fourth** duplicate value in the array and return the address where the duplicate value was found. Assign the last argument to this function the value of the duplicate. If there are no duplicate values in the array or there is not four duplicate values in the array, return a NULL value. The first occurrence of the value is not a duplicate. Look at the examples carefully; return the address of some array element in which was the fourth duplicate in the array not the fourth occurrence of the same number. (15 points) For Example,

Data set: 1 1 2 3 3 4 5 6 7 3 3 - fourth duplicate was found at element 10, the value was 3

Data set: 1 1 2 2 3 3 4 5 4 6 7 - fourth duplicate was found at element 8, the value was 4

Data set: 1 2 2 1 3 4 4 3 5 6 - fourth duplicate was found at element 7, the value was 3

Data set: 1 1 2 3 2 4 5 6 7 8 - there were not 4 duplicates in the array

Data set: 1 2 3 4 5 10 9 8 7 6 - No duplicate was found

short \*FindFourthDuplicate(short \*data, short size, short &value)

{

short \*dataEnd, \*dataFind, \*dataTest, dup;

dataFind = data + 1;

dataEnd = data + size;

for( ; dataFind < dataEnd; dataFind++)

{

for (dataTest = data; dataTest < dataFind; dataTest++)

{

if (\*dataFind == \*dataTest)

dup++;

if(dup == 4)

{

value = \*dataFind;

return dataFind + 1;

}

}

}

return NULL;

}

c) Write a void function*, CountMaximumSequence,* to find the maximum number of times any value in the data set occurred in *consecutive elements* of the array. This count must be assigned to the last argument of this function. The value of count could be 0 or some positive number depending on the data in the array. The value(s) (number) which had the maximum number of occurrences in a sequence must be assigned to the numbers array in the function. Return the number of values stored in the numbers array. (25 points)

For example: use the examples to help in solving the problem

If the data set contained: 1 2 1 3 1 4 1 5 1 6 - No value occurred in consecutive elements in the data set

If the data set contained: 1 1 2 1 3 1 4 1 5 1 1 6 – the value 1 and 1 occurred 2 times in consecutive elements

If the data set contained: 1 2 1 1 3 1 4 1 1 1 5 1 1 6 – the value 1 occurred 3 times in consecutive elements

If the data set contained: 1 2 2 1 3 1 4 1 1 5 1 1 6 – the value 2, 1 and 1 occurred 2 times in consecutive elements

If the data set contained: 1 2 1 1 1 3 4 1 1 1 1 5 1 1 6 – the value 1 occurred 4 times in consecutive elements.

short CountMaximumSequence(short \*data, short size, short \*&numbers, short &count)

{

short \*dataPtr, \*dataEnd, \*dataTest, \*dataVerify, \*numPtr, numStored;

dataPtr = data + 1;

dataVerify = data + 2;

dataTest = data;

dataEnd = data + size;

numPtr = numbers;

count = 0;

for ( ; dataPtr < dataEnd; dataPtr++, dataTest++, dataVerify++)

{

if (\*dataTest == \*dataPtr)

{

count++;

\*numPtr = \*dataPtr;

numPtr++;

}

if (\*dataTest == \*dataVerify)

//This test is to increase the ptr if the next number is the same so the previous number won't be reused

{

dataTest++;

dataPtr++;

dataVerify++;

}

}

numStored = numPtr - numbers;

return numStored;

}

**If you use the code below in the compiler, DO NOT change the code provided for you except for the two incomplete prototypes.**

/\* CSCI 152 Spring 2018 - Dr. Creider - Final exam

program will enter data into a single dimension array (duplicate values may be in arrays)

program will find the forth duplicate in the array if it exists

program will count the maximum number of times any value occurs in consecutive elements in an array

\*/

#include<fstream>

#include<iostream>

#include<stdlib.h>

#include<new>

using namespace std;

void display\_data(short \*data, short size); // function to display data in an array

short \*FindFourthDuplicate(short \*data, short size, short &value); // function to find the address of the forth duplicate value

// ############################ complete the following prototype ##################

void input\_data(/\*complete the parameter list \*/);

short CountMaximumSequence(/\*complete the parameter list \*/); // function to find max number of consecutive instances of SV

int main()

{

short \*data, size, // pointer to data array and size

count, // represents the maximum number of times any value occurred in consecutive elements

value, // value of the forth duplicate or a number found in a sequence

\*location, // address of fourth duplicate value

\*numbers, // pointer to numbers array in which to store values identifiied in the CountNaximumSequence function

Ncount; // number of elements in the numbers array assigned a value

short setNO=1; // keeps track of the number of data sets entered

cout<<"Program will store data in the array\n";

cout<<"Program will check the array for the address of the fourth duplicate value\n";

cout<<"Program will also find the maximum sequence of any value in consecutive elements\n";

cout<<"\nenter the number of values to store in the array\n or zero to terminate the program\n";

cin>>size;

while(size) // loop to permit user to test many data sets

{

input\_data(data, size, numbers); // function you define

// print the contents of the array

cout<<"\nthere are "<<size<<" values in the array - DataSet "<<setNO++<<endl;

display\_data(data, size);

// check the array for the forth duplicate value

// system("pause");

location = FindFourthDuplicate(data, size, value);

if(location) // if true, the address of the forth duplicate value was found

cout<<"The fourth duplicate was found at element number "<<location-data<<", the value was "<<value<<endl;

else

cout<<"the array did not contain any duplicates or there was not a forth duplicate value\n";

// find the number that has the maximum number of occurrences in consecutive elements

Ncount = CountMaximumSequence(data, size, numbers, count);

if(Ncount)

{

cout<<Ncount<<" numbers occurred in a maximum of "<<count<<" consecutive elements of the array\n";

display\_data(numbers, Ncount);

}

else

cout<<"No value occurred in consecutive elements in the data set\n";

delete [] data;

delete [] numbers;

//cout<<"\nenter the number of values to store in the array for a new data set\n or zero to terminate the program\n";

//cin>>size;

}

// pause the program to see the results

//system("pause");

//return 0;

}

void display\_data(short \*data, short size)

{

short i;

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

{ // display the numbers in the array separated by 1 blank all on the same line

cout<<data[i]<<' ';

}

cout<<endl<<endl;

}