Name: _key_____

PLEASE PRINT CLEARLY

ECE x70 Exam No. 1 (100pts. - 25% of the final grade)

General Remarks

This exam is limited open notes. You can have up to 3 double-side letter-size pages of notes that are stapled together and signed. Notes must not be exchanged during exam. No textbooks, computer, calculators or cell phones. Do not use back side without authorization as it may not be graded.

DL: __ ERR: __ PTS: __

DL – exam difficulty level (adjustment), ERR – exam errors, PTS – exam points.

Problem 1A (20pts.)

Consider the following function FN, and a main program that utilizes it.
Assume no pointer or other typecasting errors, and that operating system does not catch out of range memory access. Indicate what will be printed. Use "???" for unpredictable values.

```
#include <iostream>
int d = 301;
void FN(int &a, int *b, int c) {
    a = 201, *b = 202, b = (int*)203; c = 204; d = 205;
}
void main() {
    int a = 101, b = 102, c = 103, d = 104, e = 105, f = 106;
    FN(b, &c, d);
    FN(e, &f, d);
    std::cout<<a<<" "<<b<<" "<<c<<" "<<d<<" "<<e<<" "<<f<< std::endl;
} /* PRINTED:
  101      201      202      104      201      202      _____    _____
  Note: some fields may remain blank, each mistake costs 3 points until 20 */
```

Problem 1B Quick Questions (5pts.)

Answer the questions either by writing in an answer into a blank, or circling correct one from the list:

- The range of values that can be stored in `uint16_t` variable is between _0_____ and _65535_
- C/C++ language standard defines that when you reach the minimum non-zero value of IEEE standard `float` variable and multiply it by 0.5F then the new value will be:

  -1 **0** 1 same/unchanged Inf NaN/Ind undefined-by-standard

- To prevent potential compiler warning when putting `float` value into `double` variable one uses:

  nothing is needed static_cast const_cast reinterpret_cast

- The loop `for (uint32_t x=1; x>0; x++) ;` Answer: runs_forever **ends** crashes

Total errors this page: ____

Problem 2A (15pts.)

Analyze and complete the provided program. Additional clues are in variable names, comments and text printed to the screen. Some blanks may not have to be filled. – 2 points each blank

```
#include <cstddef>  /* size_t */
#include <iostream>

#include <_fstream_____________>
using namespace std;
int main() {

    _ofstream_ logFile; // to be open for writing-appending

    logFile. _open_____ ("log.txt" , _ios::app___________); // append!

    if ( logFile. _fail_____ ()) {
        cerr << "ERROR: Openign log.txt for appending failed!" << endl;
        return(1); }

    _logfile__ << "The program was run once again" << endl;

    if ( logFile. _fail_____ ()) {
        cerr << "ERROR: Appending to log.txt failed!" << endl;
        return(1); }

    logFile. _close____ ();
    return(0);
}
```

Problem 2B Quick Questions (10pts.)

Answer the following questions by circling correct one from the list: – 2 points each answer

When an exisitng file is open for writing only its content is erased:	false	**true**
When an exisitng file is open for reading only its content is erased:	**false**	true
Only text files can be open for appending:	**false**	true
In the program above we have to explicitly close the file only because of so-called good practices:	false	**true**
The same file can be open for writing, closed, and then reopened for reading in the same program:	false	**true**

Total errors this page: ____

Problem 3A (18pts.)

Implement the function that extracts the domain name from a complete Web page URL.

Hint: Assume that "http:// " is always present, you need to find the next '/' and then to look for the last '. ' before it. Plan what to do if there is no '/' or the domain name is one word without '. '. The following examples illustrate most common but not necessarily cover all possible cases. **– 2p each __**

Passed data:	return value:
"http://"	""
"http://bradley.edu/dir/page"	"edu"
"http://bradley.edu/dir/more/page"	"edu"
"http://bradley.edu/"	"edu"
"http://bradley.edu"	"edu"
"http://olek.matthewm.com.pl/bookmarks/"	"pl"
"http://olek.matthewm.com.pl/ "	"pl"
"http://olek.matthewm.com.pl "	"pl"
"http://localhost/"	"localhost"
"http://localhost"	"localhost"

```
string extension(const string & url) {
    string result="";

    if ( url._size_____() >= 7 ) {

        _size_t___ pos2 = url._find_____( _"/"_____ , 7 );

        if ( pos2 == _string::npos_ ) pos2 = url.size();

        __size_t___ pos1 = url._rfind____( _"."_____ , pos2 );

        if ( pos1 == _string::npos_ ) pos1 = 7;

        result = url._substr___( _pos1+1___, _pos2-pos1-1_ ); // pos1-1 !!
    }
    return( result );
} // Pay attention to use of ”s and ’s as they mean different data types!
```

Problem 3B Quick Questions (7pts.)

Answer the following questions by circling correct one from the list: – 1.5 points each answer

Both string::find(…) and string::rfind(…) search for substrings:	**false**	**true**
The first character of a string has the position of 1:	**false**	**true**
String::size() returns the number of characters of the string including the null character that terminates C-strings:	**false**	**true**
String::substr(3,5) returns a new string that has a size of 3:	**false**	**true**

Total errors this page: ____

Problem 4 (25pts.)

Analyze the functionality and complete blanks in the provided class definition. Use const methods when possible. Not all blanks need to be filled. **– 1.5 points each blank**

```
//* THE HEADER FILE – MovingAvg.h *

#ifndef __MYCLASS

_#define__ __MYCLASS

class MovingAvg {
private:
    const  double C1, C2; // coefficients for adjusting the moving average
    double avg;           // moving average computed as shown in updAverage(.)
    bool   grow;          // true if the average increased after most recent update
public:
    MovingAvg( _double___  _coefficient_ )

        : avg(0.0), C1(coefficient), C2(1.0-coefficient), _grow_____(true) { }

    void updAverage(double x) __________ ;

    _double___ getAverage()   _const____ { return(avg);  }

    _bool_____  getTrend()    _const____ { return(grow); }

};
#endif
--
#include "MovingAvg.h"


_void_____ _MovingAvg::_ updAverage(double x) __________ {

    _double___ new_avg = avg*C1+x*C2;

    grow =( _new_avg__ > _avg______ );

    avg = _new_avg__;

}
```

Total errors this page: ____