EE 150 Final Exams : Part I
OPEN BOOK

Monday, December 9 1996

- **Name:**
- **Student Id Number:**

Read First:

A. ****READ** THE DIRECTIONS CAREFULLY!**

B. Any materials including your text and notes may be consulted during the exam at your discretion.

C. You have a maximum of 50 minutes to complete the exam. The exam paper will be picked up from your desk (do not get up) at the end of the alloted time.

D. There are 2 problems worth a total of 50 points, the first is worth 35 points and the second is worth 15. Be careful not to spend too much time on any particular problem. **If you get stuck, go on to the next problem**. You can always come back to the one that is giving you problems.

E. Write all answers, including any scratch paper or rough notes on the pages given you.

F. If you split the exam paper, write your student id number on each and every page.

PROBLEM #1 (35 points)

As part of your new job at a software company, your are told that you must write a *function* in C which can take the address of an array of integers (the last element of which is negative 9999), and which returns a double equal to the average of the numbers in the array.

Sounds easy so far, but then you are told that you must also return the **addresses** of the minimum value in the array, the maximum value in the array, and the value in the array nearest to the average value (i.e. the median). Unfortunately your boss hates the symbols "[" and "]" and has told you to never use them in functions *you* write.

Hint 1: If you want to define a pointer to a pointer to an integer, you would say int **a. This is done in the main program below for the prototype for *processArray*. You would only need this notation in your function header.

Hint 2: A *correct* test program and output are given below:

```
#include <stdio.h>
#define ASIZE 13

main()
{
  int  A[ASIZE] = {2, 7, 1, 3, 4, 9, 5, 6, 8, 10, 55, -19, -9999};
  int *minimum;    /* minimum points to minimum value of array A */
  int *maximum;    /* maximum points to maximum value of array A */
  int *median;     /* median  points to median  value of array A */
  double average;  /* average contains the average value of array A */

  double processArray(int *, int **, int **, int **);

  average = processArray(A, &minimum, &maximum, &median);

  printf("The value of A[8]  is   8, and the address is %p\n", &A[8]);
  printf("The value of A[10] is -19, and the address is %p\n", &A[10]);
  printf("The value of A[11] is  55, and the address is %p\n\n", &A[11]);

  printf("Array Average = %5.2f\n", average);
  printf("Array Minimum = %i at addr %p\n", *minimum, minimum);
  printf("Array Maximum = %i  at addr %p\n", *maximum, maximum);
  printf("Array Median  = %i   at addr %p\n", *median,  median);
  return 0;
}
```

The output from the given test program is shown on the next page.

PROBLEM #1 (Continued ....)

The output from the given test program is shown below.

```
The value of A[8]  is   8, and the address is f7fff530
The value of A[10] is -19, and the address is f7fff538
The value of A[11] is  55, and the address is f7fff53c

Array Average =  7.83
Array Minimum = -19 at addr f7fff53c
Array Maximum = 55  at addr f7fff538
Array Median  = 8   at addr f7fff530
```

Write your solution below. Explicitly state any assumptions you make.

PROBLEM #1 (This Page for Continued Solution)

PROBLEM #1 (This Page for Continued Solution)

PROBLEM #2 (15 points)

This new job really is a pain. You have replaced a previous programmer who never wrote a comment in 10 years of C programming. You are given a program by your boss (who is still mad about your previous assignment) and told to **re-write** it so that it is more readable, shows some better style, has meaningful comments and does the same thing as it does now. He doesn't know what it does and neither does anyone else, but it is important that you don't change the way it works!

The program you are given is as follows.

```
#include<stdio.h>
main()
{
  int *L;  int *p;
  int s, i,j, ror, rand;

  scanf("%i",&s);
  if((L = (int *) malloc(s * s * sizeof(int))) == NULL) return -1;
  p = L;
  for(i=0;i<s;i++)
    for(j=0;j<s;j++) {
      scanf("%i",p);
      if ( (*p < 0) || (*p > 1) ) return -2;
      p++; }

  p = L;
  for(i=0;i<s;i++) {
      for(j=0;j<s;j++)
        printf("%i",*p++);
      printf("\n"); }

  p = L;
  for(i=0;i<s;i++) {
    ror  = 0;  rand = 1;
    for(j=0;j<s;j++) {
      ror  = ror  || *p;
      rand = rand && *p++; }
    printf("Row %i:  ROR = %i, RAND = %i\n", i, ror, rand); }
}
```

continued on next page ...

PROBLEM #2 (Continued ....)
For Problem 2,

1. First, show the output of the program given the input:

   ```
   3 0 1 1 1 0 1 1 1 1
   ```

2. And Second: re-write the program on this and the following page including better style, comments and whatever else you think would make the program easier to understand. Include a comment at the beginning of the program(s) which briefly summarizes what the program does.

PROBLEM #2 (Continued ....)