**CprE 185: Intro to Problem Solving (using C)**

**Midterm 2: Wednesday Oct 24, 2012**

**Student Name: Student ID Number:**

**Lab Section (circle one): Mon 4-6, Mon 6-8, Tue 12-2, Tue 2-4, Wed 10-12**

1. **True/False Questions (10 x 1p each = 10p)**

(a) I forgot to write down my name, student ID, and lab section. TRUE / FALSE

(b) This is a valid C statement: for(i=0,j=10;i<j;i++,j--); TRUE / FALSE

(c) The increment statement in a for loop is optional TRUE / FALSE

(d) The time function returns the seconds elapsed since 1/1/1980 GMT TRUE / FALSE

(e) The default clause of a switch statement is optional TRUE / FALSE

(f) While loops cannot be used with arrays. TRUE / FALSE

(g) In the worst case insertion sort is faster than bubble sort TRUE / FALSE

(h) A C function can take an array as an input argument TRUE / FALSE

(i) The minimum value that can be stored in an integer is -2147483648 TRUE / FALSE

(j) Linear search is faster than binary search TRUE / FALSE

1. **If-Else (5 x 2p each = 10 p)**

Given the following if/else block where a, b, c, and d are integer variables,

if(a == b || c < b ) {

d = (a + b)/2;

} else if(b == 1 || c) {

d = a + c%2;

} else

d = 2\*b;

determine the final value of the variable **d** for the following initial conditions:

a) a = 6; b = 5; c = 8; d= \_\_\_\_\_

b) a = 1; b = 1; c = 0; d= \_\_\_\_\_

c) a = 0; b = 2; c = 4; d= \_\_\_\_\_

d) a = 0; b = 1; c = 0; d= \_\_\_\_\_

e) a = 1; b =-1; c = 0; d= \_\_\_\_\_

**3. Code Snippets (2 x 5p each = 10p)**

Write a C code snippet (3-6 lines max) that produces the results specified below.

1. Print only the numbers greater than 5 and smaller than 20 that are stored in the integer array of size 10 named a. Separate the printed numbers with commas.
2. Print the numbers between 1 and 1000 that are perfect squares. In other words, they can be represented as n\*n, where n is a positive integer.

**4. What is the Output? Explain. (2 x 5p each = 10p)**

**a)**

int i=0;

for(;i>=0;i++) ;

printf("%d\n",i);

**b) Please indicate spaces with |\_\_| and new lines with \n**

int a,b;

for(a=0; a<=5; a++)

{

for(b=0; b<=5; b++)

if( (a==1) || (a==4))

printf("#");

else if((b==1) || (b==4))

printf("#");

else

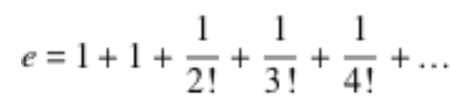
printf(" ");

printf("\n");

}

1. **Calculating e (10 p)**

The real constant **e,**  which has many applications in Mathematics and Engineering, can be calculated with the following formula:



where ‘!’ stands for factorial (N!= 1\*2\*3\* . . . \*N). Write a complete C program that approximates the value of e using the first 10 elements of the series given above.

**6. Symmetric Matrix (15 points)**

Symmetry is an interesting property in the mathematical world and particularly in matrix algebra. Your task is to determine if a given square matrix is symmetric.

As a refresher, a matrix is a 2-dimensional array of elements. A square matrix is a matrix that has the same number of rows and columns (i.e., the size is N x N). Transposition is one common operation that can be performed on a matrix. In this operation, the k-th row of the input matrix A becomes the k-th column in the output matrix B for all k=1,..,N. The square matrix A is symmetric if it is equivalent to its transpose B; that is, A = B.

The first line of the input contains the size N of the square matrix. The next N lines contain N elements each, representing the elements of the matrix. The output is simply "Symmetric" if the matrix is symmetric or "Not symmetric" if it is not.

HINT: Transposition is really just a reflection over the main diagonal of the matrix.

====== SAMPLE RUN =======

2

0 2

2 0

Symmetric

=========================

====== SAMPLE RUN =======

3

4 2 1

2 5 3

1 3 9

Symmetric

=========================

====== SAMPLE RUN =======

3

1 2 3

4 5 6

7 8 9

Not symmetric

=========================

|  |  |  |
| --- | --- | --- |
| **Question** | **Max** | **Score** |
| True/False | 10 |  |
| If-Else | 10 |  |
| Code Snippets | 10 |  |
| What is the output | 10 |  |
| Calculating e | 10 |  |
| Symmetric Matrix | 15 |  |
| Program 1 (lab) | 10 |  |
| Program 2 (lab) | 15 |  |
| Program 3 (lab) | 15 |  |
| Program 4 (lab) | 15 |  |
| Program 5 (lab) | 15 |  |
| TOTAL: | 135 |  |

May the source be with you!