CSC 111 Fall 2014
Midterm 2 Solutions

Your Name _____ **UVicID** _____

Instructions

- **Turn in your completed midterm at the front of the class and show your UVic ID.**
- **Leave through the front left door only.**
- This midterm consists of 6 pages and 12 questions.
- The marks per question are listed in square brackets for a total of 100 points.
- You have 70 minutes for this midterm. Time management—approximately 5 minutes per question.
- Attempt all questions.
- This is a closed books, closed notes, no gadgets, and no electronic devices midterm.

1. In the C programming language, which of the following functions can be used to read an entire line of text from a file with one function call? **[5]**

   ◯ `fopen()`
   ◯ `scanf()`
   ✓ `fgets()`
   ◯ `fgetc()`

2. A point in three dimensional Cartesian space consists of three coordinates: x, y, and z. Which of the following code fragments defines a syntactically correct `struct` type `Point`? **[5]**

   ◯ `structure { float x; float y; float z; } Point;`
   ✓ `typedef struct {float x, y, z; } Point;`
   ◯ `struct Point (float x, y, z; );`
   ◯ `typedef struct Point {float x, y, z}`

3. Which of the following is true? **[4]**

   ◯ Each component of a `struct` is assigned the same area of storage space.
   ◯ The syntax for `structs` is basically the same as for arrays.
   ✓ Components of `structs` may have different types.
   ◯ Each component of a `struct` must have the same type.

4. How much storage (i.e., number of bytes) is allocated on a 32 bit machine (i.e., 4 bytes are used to allocate an `int`, `float`, or pointers variable; 8 bytes for a `long` or `double` variable; 1 byte for a `bool` variable) for the following syntactically correct C array variable declarations? **[8]**

128	`double Table[4][4];`
8	`bool Vector[8];`
4000	`int* BigData[10][10][10];`
640	`typedef int Matrix [4][4]; Matrix M[10];`

5. Assuming the following declarations, write two syntactically correct C function calls to:
   [1] read from file identified by `ifp` the three dimensional coordinates of two line segments (i.e., six coordinates) into six `float` variables and
   [2] output the six `float` values read onto the console with a precision of two digits beyond the decimal point. **[10]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
   float x1, y1, z1, x2, y2, z2;
   FILE* ifp = fopen("LineSegments.txt", "r");

   fscanf(ifp, "%f %f %f %f %f %f",&x1, &y1, &z1, &x2, &y2, &z2);
   printf("%.2f %.2f %.2f %.2f %.2f %.2f\n", x1, y1, z1, x2, y2, z2);



   return EXIT_SUCCESS;
} /*main*/
```

6. Assume the following syntactically correct C declarations. Evaluate the expressions and compute the values of the Boolean variables b, c, d, and e. **[8]**

b: false

c: true

d: true

e: false

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
char* str = "CSC111";
int p = 23;
int q = 39;
int x = 49;
int y = 52;
int z = 29;
bool b = (strcmp(str, "CSC116") == 0);
bool c = (p / 7 == 3);
bool d = !(b && c);
bool e = ((p <= z && x <= q) || (y % 17 == 0));
```

7. Complete the following C function `findMin()` so that it returns the minimum value of array A? **[10]**

```
int findMin(int A[], int len) {
     int k;
     int min = A[0];

     for (k=1; k<len; k++) {
          if (A[k] < min) min = A[k];
     } /*for*/

     return min;
} /*findMin*/
```

8. Complete the following C function `shiftArrayRight()` so that it shifts all components in array A one position to the right. **[10]**

```
void shiftArrayRight(double A[], int len) {
     int k;

     double temp = A[len-1];
     for (k=len-1; k>0; k--) A[k] = A[k-1];
     A[0] = temp;

} /*shiftArrayRight*/
```

9. Complete the following C function `printUppercase()` so that it

[1] outputs all the upper case characters stored in array A on the console;

[2] counts the number of upper case letters in array A;

[3] outputs `L: 37; UC: 8` where 37 and 8 are the number of characters and uppercase letters stored in array A, respectively.

Use the `<string.h>` function `strlen()` and the `<ctype.h>` function `isupper()`. **[10]**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

void printUppercase(char A[]) {
int k;
int uc = 0;
int len = strlen(A);
for (k=0; k<len; k++) {
     if (isupper(A[k])) {
          printf("%c", A[k]);
          uc++;
     } /*if*/
} /*for*/


     printf("\n");
     printf("L: %d UC: %d\n", len, uc);
} /*printUppercase*/

int main(void) {
     printUppercase("Once Upon A Time There Was Polar Bear");
     return EXIT_SUCCESS;
} /*main*/
```

Here is the output produced by the following function call:

```
     printUppercase("Once Upon A Time There Was Polar Bear");
```

OUATTWPB
L: 37; UC: 8

10. Consider the following syntactically correct C program called `alice.c`. Describe the effect and the exact output of this program when executing this program. **[10]**

```
#include <stdio.h>
#include <stdlib.h>
#define LINEMAX (300)
int main(void) {
      char line[LINEMAX];
      FILE *ifp = fopen("alice.c", "r");
      if (ifp == NULL) exit(EXIT_FAILURE);
      int n = 0;
      while(!feof(ifp)) {
            if (fgets(line, LINEMAX, ifp)) {
               if (n<3) printf(">>>%s", line);
               n++;
            } /*if*/
      } /*while*/
      printf("n = %d\n", n);
      fclose(ifp);
      return EXIT_SUCCESS;
} /*main*/
```

```
The program outputs itself, but only the first three lines.
It also counts the number of lines in the program.
The output is as follows:
>>>#include <stdio.h>
>>>#include <stdlib.h>
>>>#define LINEMAX (300)
n = 18
```

11. What is the output of the following syntactically correct C program? **[10]**

```
#include <stdio.h>
#include <stdlib.h>
#define AMAX (27)
int main(void) {
   char alphabet[AMAX];
   int k;
   char ch = 'A';
   for (k=0; k<=AMAX-1; k++) { alphabet[k] = ch; ch++; }
   alphabet[26] = '\0';
   printf("%s\n", alphabet);
   return EXIT_SUCCESS;
} /*main*/
```

```
      ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

12. Consider the following syntactically correct C program. **[10]**

```
#include <stdio.h>
#include <stdlib.h>
#define MAX (15)

typedef int Index;
typedef int Item;

void initRand(Item A[], Index len) {
     Index k;
     for (k=0; k<len; k++) { A[k] = rand() % len; }
} /*initRand*/

void printArray(Item A[], Index len) {
   Index k;
   for (k=0; k<len; k++){ printf("%d ", A[k]); }
   printf("\n");
} /*printArray*/

int main(void) {
   Item A[MAX];
   initRand(A, MAX);
   printArray(A, MAX);
   return EXIT_SUCCESS;
} /*main*/
```

7	How many variable declarations are in this C program?
3	How many function declarations are in this C program?
5	How many function calls are in this C program?
2	How many type declarations are in this C program?
3	How many preprocessor directives are in this C program?