CSC 111 Fall 2013
Midterm 2 Solutions

Your Name | Müller / Miller | **UVicID** | A01/A02

Instructions

- This midterm consists of 4 double-sided pages and 17 questions.
- The questions are worth 4, 6, or 8 points for a total of 100 points. The points are given in square brackets at the end of each question.
- You have 70 minutes for this midterm. **Time management—approx. 4 minutes per question.**
- Attempt all questions.
- This midterm is closed-books, closed-notes, no calculators, no gadgets, and no electronic devices.
- Turn in your completed midterm in **at the front of the class. Show your UVic ID Card.**
- **Do not leave before 45 minutes after the start of the midterm.**

1. A complex number consists of two parts: a real (**re**) and an imaginary (**im**) part. Which of the following code fragments defines a syntactically correct `struct` type `Complex`? **[4]**

    ○   `structure { double re; double im; } Complex;`

    ❖   `typedef struct {double re, im; } Complex;`

    ○   `typedef struct Complex {double re, im }`

    ○   `struct Complex (double re, im; );`

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

    ○   Each component of a `struct` is assigned the same area of storage space.

    ○   The syntax for `struct`s is basically the same as for arrays.

    ○   Each component of a `struct` must have the same type.

    ❖   Components of `struct`s can have different types.

3. In the C programming language, how do you refer to a file when you read, write or close a file? **[4]**

    ○   fopen()

    ○   printf()

    ❖   FILE* pointer

    ○   fgetc()

4. Consider the following syntactically correct C program called `reflection.c`. What is the effect when
    you execute this program? **[6]**

```c
#include<stdio.h>
#include<stdlib.h>
#define MAX (300)
#define FNAME ("reflection.c")
int main(void) {
        char line[MAX];
        FILE *ifp = fopen(FNAME, "r");
        if (ifp == NULL) {
                printf("Input file %s not found\n", FNAME);
                exit(EXIT_FAILURE);
        } /*if*/
        int n = 0;
        while(!feof(ifp)) {
                if (fgets(line, MAX, ifp)) n++;
        } /*while*/
        printf("n = %d\n", n);
        fclose(ifp);
        return EXIT_SUCCESS;
} /*main*/
```

⭘   This program will create a new file.

⭘   This program will output the program text of this program.

❖   This program will count the number of lines in its source file `reflection.c`

⭘   This program will copy file `reflection.c` to standard output.

5. Consider the following syntactically correct C declarations and assignments. **[6]**

```c
int a;
int *b;
int **c;
a = 17;
b = &a;
c = &b;
```

Write one syntactically correct C call to `printf()` to output the addresses of variables `a` and `b`.

```c
printf("address of a = %p   address of b = %p\n", b, c);
printf("address of a = %p   address of b = %p\n", &a, &b);
printf("address of a = %p   address of b = %p\n", &a, c);
printf("address of a = %p   address of b = %p\n", b, &b);
```

6. Consider the following syntactically correct C declarations: **[6]**

```
#include <string.h>
#define MAX_SIZE (300)

typedef struct {
      char first[MAX_SIZE];
      char last[MAX_SIZE];
      float mark;
} Person;
Person student;
```

Initialize variable student with your `first` and `last` name as well as the `mark` you hope to get on this midterm. **Hint:** Use a function defined in the C standard library `<string.h>` such as `strlen(str), strcmp(str1, str2), strcpy(dest, source), or strcat(dest, source)`.

```
      strcpy(student.first, "Michael");
      strcpy(student.last, "Miller");
      student.mark = 100;
```

7. Assume the following syntactically correct C declarations. **[4]**

```
#define MAX_SIZE (300)
char str[MAX_SIZE];
```

Describe the difference between the following two syntactically correct C statements.

a) `scanf("%s", str);`
b) `fgets(str, MAX_SIZE, stdin);`

```
fgets() returns an entire line including white space and '\n'
scanf() only returns until the string before the first white space
both return a proper C string with '\0' to terminate the string
```

8. Which one of these declarations is a syntactically correct C type declaration? **[4]**

○   `typedef Item float;`

○   `typedef index int;`

❖   `typedef int boolean;`

○   `typedef float Vector[100, 100];`

9. What is the output of the following syntactically correct C program? **[8]**

```
#include <stdio.h>
#include <stdlib.h>
#define MIN (-4)
#define MAX (4)
#define MOD (3)
int main(void) {
      int k, z;
      for (k=MAX;  k > MIN;  k--) {
            z = k % MOD;
            printf("%d ",  z);
      } /*for*/
      printf("\n");
      return EXIT_SUCCESS;
} /*main*/
```

Output: `1 0 2 1 0 -1 -2 0`

10. Assume the following syntactically correct C declarations. Which one of the following Boolean expressions evaluates to false or 0? **[4]**

```
#include <stdbool.h>
int x = 17;
bool a = true;
bool b = !false;
bool c = true;
bool d = (99 != x);
```

○   c || d

○   a && b || !d

○   !(a && b || c)

○   x == 17

11. Write a syntactically correct C function called `swap()` to exchange the values of two `float` values accessed via parameters. **[8]**

```
true

true

false

true
```

12. What is the console output of the following syntactically correct C program? **[6]**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define    OUTPUT     (" Once upon a time there was a polar\n")

int main(void){
     char str[] = OUTPUT;
     FILE *ifp;
     FILE *ofp;

     ofp = fopen("csc111.txt", "w");
     fputs("Programming is really cool!\n", ofp);
     fclose(ofp);
     ifp = fopen("csc111.txt", "r");
     while(!feof(ifp)) {
          if (fgets(str, strlen(str), ifp) != NULL) {
               printf("%s", str);
          }/*if*/
     }/*while*/
     printf("My favorite course is CSC 111!\n");
     fclose(ifp);
     return EXIT_SUCCESS;
} /*main*/
```

○   Once upon a time there was a polar bear
      Programming is really cool!
      My favorite course is CSC 111!

○   My favorite course is CSC 111!

❖   Programming is really cool!
      My favorite course is CSC 111!

○   Programming is really cool!

13. What is the output of the following syntactically correct C program? **[8]**

```
#include <stdio.h>
#include <stdlib.h>
#define  VSize  (4)
typedef int Index;
typedef float Item;
typedef Item Vector[VSize];

void initVector(Vector V, Index size, Item z) {
    Index k;
    for (k=0; k<size; k++) V[k] = (Item)(k)*z;
} /*initVector*/

void printVector(const Vector V, Index size) {
    Index k;
    for (k=0; k<size; k++) printf("%.1f  ", V[k]);
} /*printVector*/

int main(void) {
    Vector Vec;
    initVector(Vec, VSize, 5.0);
    printVector(Vec, VSize);
    initVector(Vec, VSize, 7.0);
    printVector(Vec, VSize);
    printf("\n");
    return EXIT_SUCCESS;
} /*main*/
```

○   0   7   14   21   0   5   10   15

○   0.0 7.0 14.0 21.0 0.0 5.0 10.0 15.0

○   0   5   10   15   0   7   14   21

❖   0.0 5.0 10.0 15.0 0.0 7.0 14.0 21.0

14. What is the output of the following syntactically correct C program? **[8]**

```
#include <stdio.h>
#include <stdlib.h>
#define VSIZE  (4)
typedef float Vector[VSIZE];
void func1(Vector a, int len) {
     int k; float first = a[0];
     for (k=0; k<len-1; k++)  a[k] = a[k+1];
     a[len-1] = first;
} /*func1*/
void func2(Vector a, int len) {
     int k;
     for (k=0; k<len; k++)  printf("%.1f  ", a[k]);
     printf("\n");
} /*func2*/
int main(void) {
     Vector vec;
     vec[0] = 1.1; vec[1] = 5.5; vec[2] = 4.4; vec[3] = 3.3;
     func1(vec, VSIZE); func1(vec, VSIZE); func2(vec, VSIZE);
     return EXIT_SUCCESS;
} /*main*/
```

O   4.4   3.3   5.5   4.4

O   5.5   4.4   3.3   1.1

O   1.1   5.5   4.4   3.3

❖   4.4   3.3   1.1   5.5

15. What is the output of the following syntactically correct C program? **[6]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int z = 0; int n = 13; int k = 1;
     int* p = &k;
     while (*p <= n) {
          z = z + *p;
          *p = *p + 2;
     } /*while*/
     printf("n = %d  z = %d\n", n, z);
     return EXIT_SUCCESS;
} /*main*/
```

O   n = 13 z = 72

O   n = 13 z = 36

O   n = 13 z = 66

❖   n = 13 z = 49

16. Assume the following syntactically correct C code. Write a loop that fills the array with the following repeated character sequence <->. Make sure you don't write past the end of the array. **Hint:** Use a `for` loop with `if` statements inside the `for` loop. **[8]**

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX (200)
int main(void) {
    char buffer[MAX];
    int k;

    for (k=0; k<MAX; k++) {
        if (k%3 == 0) buffer[k] = '<';
        if (k%3 == 1) buffer[k] = '-';
        if (k%3 == 2) buffer[k] = '>';
    } /*for*/

    for (k=0; k<MAX; k++) printf("%c", buffer[k]);
    return EXIT_SUCCESS;
} /*main*/
```

17. What is the output of the following syntactically correct C program? **[6]**

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int day; int month; int year;
} Date;
void initDate(Date* d) {
    d->day = 19; d->month = 11; d->year = 1999;
} /*initDate*/
int printDate(Date d) {
    printf("Date: %d/%d/%d\n", d.day, d.month, d.year);
    return EXIT_SUCCESS;
} /*printDate*/
int main(void) {
    Date bd = {99, 99, 99};
    printDate(bd); initDate(&bd); printDate(bd);
    return EXIT_SUCCESS;
} /*main*/
```

```
Date: 99/99/99
Date: 19/11/1999
```