University of Victoria
Final Examination Solutions
December 2013

Last Name:	**Müller & Miller**
First Name:	
UVic Student Number:	**V00**

Course Name & Number	CSC 111 Fundaments of Programming with Engineering Applications	
Sections	A01	A02
CRN	10711	10712
Instructors	H. A. Müller	D. M. Miller
Duration	3 hours	

Instructions

- This examination consists of 11 pages including this cover page. Count the number of pages and report any discrepancy immediately to an invigilator.
- Answer the questions on the examination paper.
- There are 24 questions. The questions are each worth 2, 3, 4, or 6 points, for a total of 100 points. The points are listed in square brackets at the end of the first line of each question.
- You have 3 hours for this examination. **Time management: approx. 5-7 minutes per question.**
- This examination is closed-books, closed-notes, no calculators, no gadgets, and no electronic devices. **Cell phones must be turned off**.
- Turn in your completed final exam at the front of the examination room.
- For multiple choice questions, mark all circles that are next to correct choices.
- Be sure to complete the information on the declaration attached to this examination **including your signature.** Do not detach the declaration.
- **You are not permitted to leave before 3:30 p.m.**

1. Which of the following code fragments correctly defines a structure type `Complex` in the programming language C? **[2]**

   ○ 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 not a characteristic of the C programming language? **[2]**

   ○ Rich operator set

   ○ Automatic garbage collection

   ○ Famous I/O library

   ○ Ideal for systems programming

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

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void){
     char s[] = "This Land--From Bonavista to Vancouver Island";
     FILE* ifp;
     FILE* ofp;
     ofp = fopen("csc111.txt", "w");
     fputs("Listening to the wind of change", ofp);
     putc((int)('\n'), ofp);
     fclose(ofp);
     ifp = fopen("csc111.txt", "r");
     while(fgets(s, strlen(s), ifp) != NULL)  printf("%s", s);
     fclose(ifp);
     strcat(s, "---Scorpions");
     ofp = fopen("csc111.txt", "w");
     fputs(s, ofp);
     fclose(ofp);
     ifp = fopen("csc111.txt", "r");
     while(fgets(s, strlen(s), ifp) != NULL)  printf("%s", s);
     fclose(ifp);
     return EXIT_SUCCESS;
} /*main*/
```

```
Listening to the wind of change
Listening to the wind of change
---Scorpions
```

4. Which of the following declarations is syntactically correct and will allocate memory to store an integer and initialize a variable p to point to the allocated storage area? **[2]**

○ `int p = (int) malloc( sizeof(int) );`

[✓ ○ `int* p = (int*) malloc( sizeof(int) );`]

○ `int* p = (int) malloc( sizeof(int *) );`

○ `int* p = (int*) malloc(int);`

5. Give a single syntactically correct C expression that will evaluate to `true` if an integer variable x is divisible by 3, 5 or 7. **[3]**

Boolean expression: x%3==0 || x%5==0 || x%7==0

6. Which of the following operators has the highest precedence in the programming language C? **[2]**

○ +

○ ==

○ ++

[✓ ○ ->]

7. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
void magic(int* c, int d) {
     printf("%d ", *c);     printf("%d ", d);
     *c = *c * 3 + d;
     printf("%d ", d);     printf("%d ", *c);
} /* magicWand */
int main(void) {
     int x = 21;     int y = 19;
     printf ("%d ", x);     printf ("%d ", y);
     magic (&x, y);
     printf ("%d ", x);     printf ("%d\n ", y);
     return EXIT_SUCCESS;
} /* main */
```

○ 21 19 21 19 82 19 82 19

[✓ ○ 21 19 21 19 19 82 82 19]

○ 21 19 21 19 103 19 103 19

○ 21 19 21 19 82 19 21 19

○ None of the above

8. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int x = 19;
     int a = 1;
     x  -=  3  * x  +  ( a = 18 );
     printf("%d\n", x);
     return EXIT_SUCCESS;
} /*main*/
```

○ -76

○ 56

○ -38

● -56

○ None of the above

9. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
   int k = 9;
   while (k > 8) {
        if (k % 2) printf("%d ", k);
        k = k + k/2;
        if (k > 27) break;
   } /*while*/
   for (k=5;  k>6;  k=k-1) printf("%3d",  k);
   printf("\n");
   return EXIT_SUCCESS;
} /*main*/
```

○ 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

○ 9 11 13 15 17 19 21 23 25 27

○ 9 11 13 15 17 19

○ 9 13 18

● None of the above

10. Circle each statement in the following C program that could cause a memory access error. **[6]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
   int a, b, c;
   int* x;
   int* y;
   int** z;
   a = 17;
   *y = b;
   *z = x;
   b = *y;
   c = a*b;
   printf("c = %d\n", c);
   return EXIT_SUCCESS;
} /*main*/
```

11. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int z = 0, n = 11, 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*/
```

○ n = 11 z = 16

○ n = 11 z = 36

[✓ ○ n = 11 z = 25]

○ n = 11 z = 47

○ None of the above

12. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
void printBool(bool b) {
     if (b) printf("true "); else printf("false ");
} /*printBool*/
int main(void) {
     int a, *b, *c, d;
     a = 5; b = &a; c = b;
     bool b1 = (b == c);
     printBool(b1);
     bool b2 = (&a == c);
     printBool(b2);
     bool b3 = (b == &d);
     printBool(b3);
     bool b4 = (*b == 5);
     printBool(b4);
     printf("\n");
     return EXIT_SUCCESS;
} /*main*/
```

```
true true false true
```

13. What is the console output of the following syntactically correct C program? **[4]**

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int k = 1;
     while (k < 12) {
            printf("%d ",  k%7);
            k = k + 2;
     } /*while*/
     printf("\n");
     return EXIT_SUCCESS;
} /*main*/
```

```
1 3 5 0 2 4
```

14. Consider the following C declarations: **[3]**

```
#include <string.h>
typedef struct {
   int year;
   char month[11];
   int day;
} Date;
Date dob;
Date *d = &dob;
```

Using variable d initialize dob with the birthday May 29, 1917.

```
strcpy(d->month, "May");

d->day = 29;

d->year = 1917;
```

15. Write a syntactically correct C program that prompts the user to enter an integer value n, read the value from stdin, and generate the first n values in the sequence given below with all values printed on a single line: **[6]**

```
1 3 6 10 15 21 28 36 45 55 …
```

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int n = 0;
     int k;
     int s = 0;
     printf("Enter n: ");
     fflush(stdout);
     scanf("%d", &n);
     for(k=1; k<=n; k++){
          s += k;
          printf("%d ", s);
     } /*for*/
     printf("\n");




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

16. Complete the following C function so that it is syntactically correct and behaves as described in the comments. **[6]**

```
void rotateFloatArray(float x[], int n){
     // rotates an array of floats so that the value
     // in each position p, 1 <= p <= n-1, is moved to
     // position p-1 and the value in position 0 is
     // moved to position n-1.
     float temp;
     int k;
     temp = x[0];
     for (k=0; k<n-1; k++) x[k] = x[k+1];
     x[n-1] = temp;

} /*rotateFloatArray*/
```

17. What is a binary tree? **[2]**

○ A special case of a tree that stores only binary values (i.e., 0’s and 1’s).

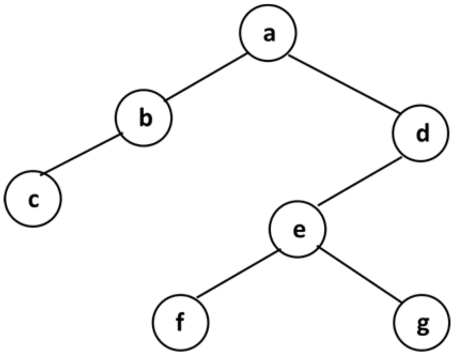
○ A special case of a tree where each node has two children.

○ A forest consisting of 2 trees.

○ A data structure where nodes are linked to at least two other nodes.

○ None of the above.

18. Consider the binary tree shown below. In what order are the nodes a through g visited using the `preorder`, `inorder` and `postorder` binary tree traversal algorithms? **[6]**



preorder

a b c d e f g

inorder

c b a f e g d

postorder

c b f g e d a

19. What is the difference between `call-by-value` and `call-by-reference` parameter passing in the programming language C? Explain the differences in your own words. **[6]**

- The main parameter passing mechanism in C is call-by-value
- Call-by-reference has to be simulated using pointers in C
- Arrays are passed using call-by-reference in C
- Most languages explicitly support call-by-value and call-by-reference
- Fortran: call-by-reference and by value-result
  Pascal, C++:, Java, C# call-by-reference and call-by-value

- **Def. Call-by-value.** Variables that are passed to a function using call-by-value cannot be changed by the function.
  Call-by-value parameters are said to be *input only.*
- **Def. Call-by-reference.** Variables that are passed to a function using call-by-reference can be changed by the function. Call-by-reference parameters are said to be
  input and output.

20. Which data structure can be represented with the following C declarations? **[2]**

```
typedef struct { float info; } Item;
typedef struct Item*  ItemRef;
typedef struct NodeStruct*  NodeRef;
typedef struct NodeStruct {
     ItemRef  item;
     NodeRef  right;
} Node;
```

○ Graph

○ Doubly linked list

○ Binary tree

○ [✓ Singly linked list]

○ None of the above

21. Describe the console output of the following syntactically correct C program? Note you do not have to provide the actual output—just describe it. **[6]**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define LOW_RANGE  (25)
#define HIGH_RANGE (39)
#define MAX_NUM (8)
int main(void) {
   int rn = 0;
   unsigned int seed = (unsigned int)time(NULL);
   srand(seed);
   int k = 0;
   while (k < MAX_NUM) {
        do
             rn = rand();
        while (rn < LOW_RANGE || HIGH_RANGE < rn);
        printf("%d: %d\n", k, rn);
        k = k + 1;
   } /*while*/
   return EXIT_SUCCESS;
} /*main*/
```

```
0: 34
1: 25
2: 33
3: 31           Sample output
4: 36
5: 37
6: 26
7: 25
```

The numbers in the second column are uniformly distributed random numbers in the range of 25 to 39.

22. What is the difference between a `stack` and a `queue` data structure? **[4]**

`Stack`: `Insert` (i.e., `push`) and `delete` (i.e., `pop`) at the same end of a list.

`Queue`: `Insert` (i.e., `enqueue`) and `delete` (i.e., `dequeue`) at different ends of a list.

23. Write a syntactically correct C function named `range()` that returns the range of values in an unsorted `double` array. For example, the range of values in the following `double` array is 5.3. **[6]**

3.1	4.2	4.6	5.2	6.1	7.5	8.4	7.0	4.9	6.6

```
double range(double a[], int n){
   double min = a[0];
   double max = a[0];
   int k;
   for (k=1; k<n; k++){
         if (a[k] < min) min = a[k];
         if (a[k] > max) max = a[k];
   } /*for*/
   return max-min;







} /*range*/
```

24. The most frequent letter in English is the letter 'E'. Write a syntactically correct C function called `eCount()` which accepts a pointer to a '\0' terminated string of characters as a parameter and computes and returns the number of occurrences of the characters 'e' and 'E' in the string. **[6]**

```
#include <ctype.h>
int eCount(char *s){
     int eCnt = 0;
     while (*s != '\0') {
          if (*s=='e'||*s=='E')  eCnt++;
          s++;
     } /*while*/
     return eCnt;







} /*eCount*/
```

***** END of EXAMINATION *****