University of Victoria
Final Examination
December 2014

Last Name:	
First Name:	
UVic Student Number:	**V00**

Course Name & Number	CSC 111 Fundaments of Programming with Engineering Applications
Sections	A01 & A02
CRN	10691 & 10692
Instructors	H. A. Müller
Duration	3 hours

Instructions

- This examination consists of 12 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, 5, 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, no cell phones, and no electronic devices. **Cell phones must in your bag and 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.**

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

- ◯ `structure { double re; double im; } Complex;`
- ◯ `struct Complex (double re, im; );`
- ◯ `typedef struct {double re, im; } Complex;`
- ◯ `typedef struct Complex {double re, im }`

!. Consider the following syntactically correct C program called `wonderland.c`. What is the exact output when executing this program. **[6]**

```
#include <stdio.h>
#include <stdlib.h>
#define LINECHARMAX (400)
#define LINEMAX (5)
int main(void) {
     printf("Hello Alice\n");
     char line[LINECHARMAX];
     FILE *ifp = fopen("wonderland.c", "r");
     if (ifp == NULL) exit(EXIT_FAILURE);
     int n = 0;
     while(!feof(ifp)) {
          if (fgets(line, LINECHARMAX, ifp)) {
             if (n < LINEMAX) printf("###%s", line);
             n++;
          } /*if*/
     } /*while*/
     printf("n = %d\n", n);
     fclose(ifp);
     return EXIT_SUCCESS;
} /*main*/
```

3. Assume the following syntactically correct C declarations. Evaluate the expressions and compute the values of the Boolean variables b, c, d, and e. **[4]**

b1 ☐

b2 ☐

b3 ☐

b4 ☐

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
char* str = "CSC225";
int p = 38;
int q = 21;
int x = 49;
int y = 51;
int z = 61;
bool b1 = (strcmp(str, "CSC226") == 0);
bool b2 = (p % 7 == 3);
bool b3 = !(b1 || b2);
bool b4 = ((p <= z && x <= q) || (y % 17 == 0));
```

4. Complete the following C function `findMin2()` so that it returns the second smallest value of array `A` where `len` is the number of array components? **[6]**

```
int findMin2(int A[], int len) {
     /* assume len >= 2 */
     int k;
     int min = A[0];  int min2 = A[1];
```

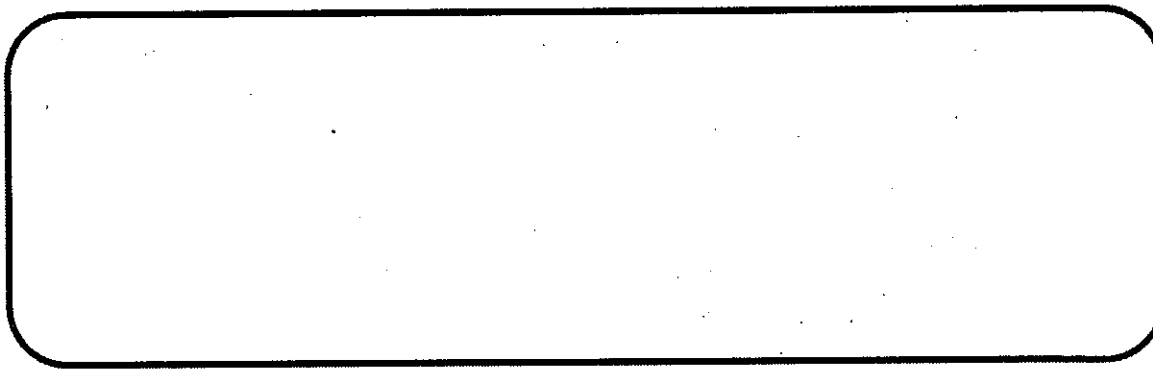
```
     return min2;
} /*findMin2*/
```

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

- ◯ `double* p = (double) malloc( sizeof(double *) );`
- ◯ `float* p = (float*) malloc(float);`
- ◯ `double* p = (double*) malloc( sizeof(double) );`
- ◯ `float p = (float) malloc( sizeof(float) );`

6. Complete the following C function `shiftArrayRight()` so that it shifts all components in array `A` one position to the right where `len` is the number of array components. **[6]**

```
void shiftArrayRight(double A[], int len) {
     /* assume len >= 2 */
     int k;
```



```
} /*shiftArrayRight*/
```

7. Which of the following is not true for the programming language C? **[2]**

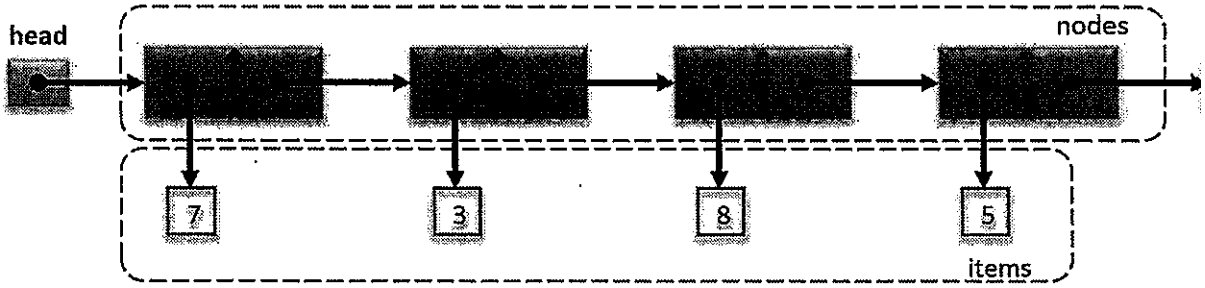
- ◯ The components of an array must be of the same type.
- ◯ Any `for` loop can be converted to a `while` loop.
- ◯ Components of `structs` may have different types.
- ◯ A proper string is terminated with the `'\0'` character.
- ◯ `int, double, float, string, and bool` are basic data types.
- ◯ `structs` and arrays may be nested.
- ◯ Assignments are expressions.

8. Give a single syntactically correct C expression that will evaluate to `true` if an integer variable `x` is divisible 13, 17, or 19. **[4]**

Boolean expression:

9. Assume that the singly linked list structure depicted below was created using the following type and variable declarations.

```
typedef  int Info;
typedef  struct {
   Info info;
} Item;
typedef  Item* ItemRef;
typedef  struct NodeStruct* NodeRef;
typedef  struct NodeStruct  {
     ItemRef item;
     NodeRef next;
} Node;
NodeRef head;
```



a) Using one assignment change the value of `item 8` to a value of `4`. **[3]**

b) Using one assignment disconnect the second part of the list (i.e., the nodes with item 4 and 5) so that they cannot be reached using `head`. **[3]**

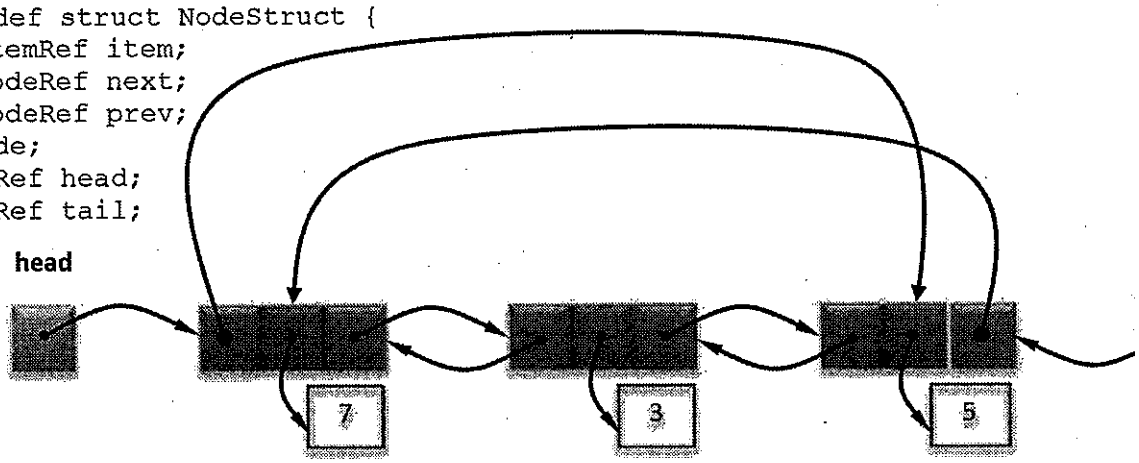
10. Which of the following operators does not have highest precedence in the programming language C? **[2]**

- ◯ ()
- ◯ []
- ◯ &
- ◯ ->

1. Assume that the circular doubly linked list structure depicted below was created using the following type and variable declarations.

```
typedef int Info;
typedef struct {
   Info info;
} Item;
typedef Item* ItemRef;

typedef struct NodeStruct* NodeRef;
typedef struct NodeStruct {
   ItemRef item;
   NodeRef next;
   NodeRef prev;
} Node;
NodeRef head;
NodeRef tail;
```



a) Give two different ways of changing the value of `item 3` to a value of `4` (i.e., give two different assignments). **[3]**

b) Describe the effect of the following assignment. **[3]**

```
tail->next->next->prev->prev->next->prev->prev->prev->item->info = 6;
```

12. 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 data structure consisting of 2 trees which is also called forest.
- ◯ A data structure where nodes are linked to at least two other nodes.
- ◯ A special case of a tree where each node has at most two children.
- ◯ None of the above.

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

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

void valPusterla(int* c, int d) {
     printf("%d ", *c);     printf("%d ", d);
     *c = *c * 3 + d;
     printf("%d ", d);     printf("%d ", *c);
} /* valPusterla */

int main(void) {
     int x = 35;     int y = 37;
     printf ("%d ", x);     printf ("%d ", y);
     valPusterla(&x, y);
     printf ("%d ", x);     printf ("%d\n ", y);
     return EXIT_SUCCESS;
} /*main*/
```

- ◯ 35 37 35 37 42 37 82 37
- ◯ 35 37 35 37 37 142 142 37
- ◯ 35 37 35 37 142 37 142 37
- ◯ 35 37 35 37 142 37 35 37
- ◯ None of the above

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

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#define AMAX (30)
int main(void) {
   char alpha[AMAX];
   int k;
   char ch = 'A';
   for (k=0; k<=AMAX-1; k++) {
        if (k % 2 == 0) alpha[k] = tolower(ch); else alpha[k] = ch;
        ch++;
   } /*for*/
   alpha[26] = '\0';
   printf("%s\n", alpha);
   return EXIT_SUCCESS;
} /*main*/
```

5. Write a syntactically correct C function `print2DArray()` that out puts the values of a two dimensional `double` array `A` with `maxrows` rows and `maxcols` columns. **[6]**

0.0	1.0
1.0	2.0
2.0	3.0
3.0	4.0

```
void print2DArray(double A,  int maxrows, int maxcols) {
```

```
} /*print2DArray*/
```

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

```
#include <stdio.h>
#include <stdlib.h>'
int main(void) {
     int x = 17;
     int a = 2;
     x  -=  3  * x  +  ( a = 19 );
     printf("%d\n", x);
     return EXIT_SUCCESS;
} /*main*/
```

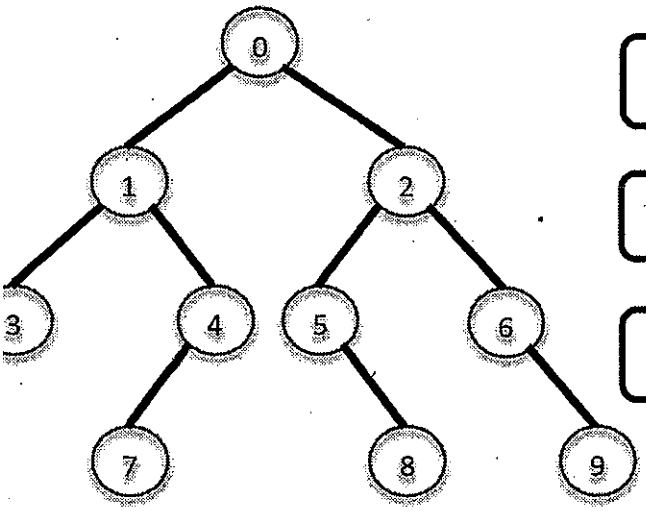
- ◯ -79
- ◯ 53
- ◯ -31
- ◯ -53
- ◯ None of the above

17. Consider the following C declarations: **[2]**

```
#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 `August 27, 1925.`

18. Consider the binary tree shown below. In what order are the nodes 0 through 9 visited using the `preorder`, `inorder` and `postorder` binary tree traversal algorithms? **[6]**



preorder

inorder

postorder

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. Consider the following data structure traversal pattern. Which of the following data structures can be traversed effectively using this pattern? **[2]**

```
NodeRef x = tail;
while (x != head) {
    // process list element
    x = x->prev;
} /*while*/
```

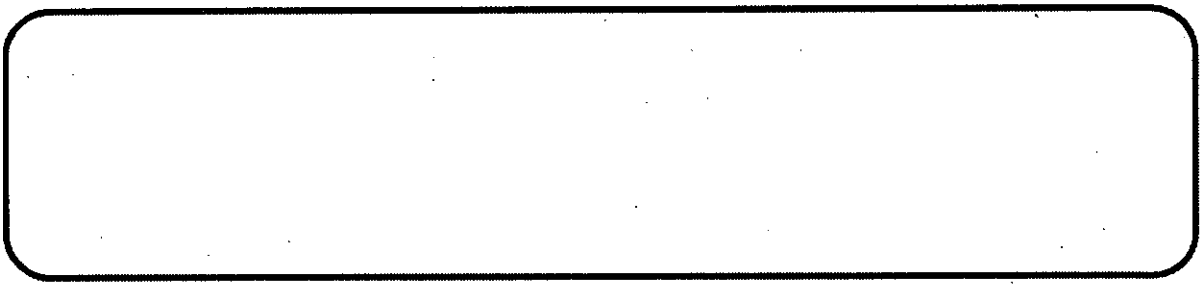
- ◯ Singly linked list
- ◯ Singly linked circular list
- ◯ Doubly linked circular list
- ◯ Binary tree
- ◯ None of the above

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

22. 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]**

3. Write a syntactically correct C function that swaps two characters in a `char` array called `word`, where `rn1` and `rn2` are the array indexes of the two characters to be swapped. **[5]**

```
void swap(char word[], int rn1, int rn2) {
```



```
} /*swap*/
```

4. How would you describe the console output of the following syntactically correct C program in detail (e.g., provide sample output)? **[5]**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define LOW_RANGE  (31)
#define HIGH_RANGE (51)
#define MAX_NUM (11)
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*/
```

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