83/90

CSC 111 Fall 2013
Midterm 1

Your Name [redacted] **UVicID** [redacted]

Instructions

- This midterm consists of 5 pages and 11 questions.
- The first 5 questions are worth 6 points each for a total of 30 points. The last 6 questions are worth 10 points each for a total of 60 points. The complete midterm is worth 90 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.
- **Turn in your completed midterm at the front of the class and show your UVic ID.**
- Leave through the front door.

1. What are the values of the following C expressions?
   Assume the following C declarations and initializations:

```
int x = 3;
int a = 1;
```

`(x != 3)`	x ≠ 3 (false)
`(0>x || x>9)`	false
`(a += 2*(a - 19))`	-35
`((x*17) % 2)`	1

6 ✓ ✓ ✓

2. For the following statements, check the correct circle.
   The C preprocessor

- ○ compiles C programs
- ○ builds an application or an executable
- ○ checks for semantic errors
- ⊗ includes text files using #include and substitutes text using #define directives

6

12

3. Consider the following syntactically correct C program.

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    printf("CSC 111\n");
    return EXIT_SUCCESS;
}
```

How many function names appear in this program? Check the correct circle.

6

- ○ 0
- ○ 1
- ⊗ 2
- ○ more than 2

printf is a function defined in the stdio.h header of the language C

4. How many formal parameters are there in this function header `int f(int s, float x)`

6

- ○ 0
- ○ 1
- ⊗ 2
- ○ more than 2

5. Given the following two declarations and initializations, how do you store the value 17 into the integer variable x using pointer p? Check the correct circle.

```
int x = 3;
int* p = &x;
```

6

- ○ `p* = 17;`
- ○ `x = 17;`
- ⊗ `*p = 17;`
- ○ `*p* = 17;`

18

6. How many proper identifiers are in each line according to C syntax? Enter the number for each line in the circle at the front of the line.

(6) ABC aBc _systematic B777 k 711 int mega %d

(4) These/* are */ identifiers with a //caveat

(6) 17.4 nine 19.6 forty-four hexadecimal k modulo output

(9) Check for C C++ C# comments keywords such as _for while long

7. Given the following C declarations and initializations, create four pointer variables to point to these four variables.

```
int k = 17;
char c = 'A';
float f = 3.14;
double d = 2.81;
```

int * kpoint = &k;

char* cpoint = &c;

float *fpoint = &f;

double* dpoint = &d;

8. What is the output of the following syntactically correct C program?

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
   int k = 77;
   while (k > 47) {
        printf("%d  ", k);
        k = k - 10;
   }/*while*/
   printf("Finished\n");
   return EXIT_SUCCESS;
} /* main */
```

Output:

77 67 57 Finished

(cursor on new line, below the line that was just printed

(it is not visible unless you put something else)

26

9. What is the output of the following syntactically correct C program?

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
     int k = 21;
     while (k < 35) {
          if (k % 2 == 0)
               printf("%d\n", k);
          k = k + 3;
     }/*while*/
     printf("Common sense!\n");
     return EXIT_SUCCESS;
} /* main */
```

Output:

24
30
Common sense!

10

21, 24, 27, 30, 33,

cursor goes to new line after printing last character

10. What is the output of the following syntactically correct C program?

(not visible, unless you print something else)

```
#include <stdio.h>
#include <stdlib.h>
/* function prototypes */
int main(void);
void f1(void);
void f2(void);
void f3(void);
void f4(void);

void f1() { printf("f1  "); f2(); }
void f2() { printf("f2  "); f4(); }
void f3() { printf("f3  "); f4(); }
void f4() { printf("f4  "); }
int main(void){
   printf("main ");
   f1();
   f3();
   printf("Bye\n");
   return EXIT_SUCCESS;
}
```

Output:

main f1 f2 f4 f3 f4 Bye

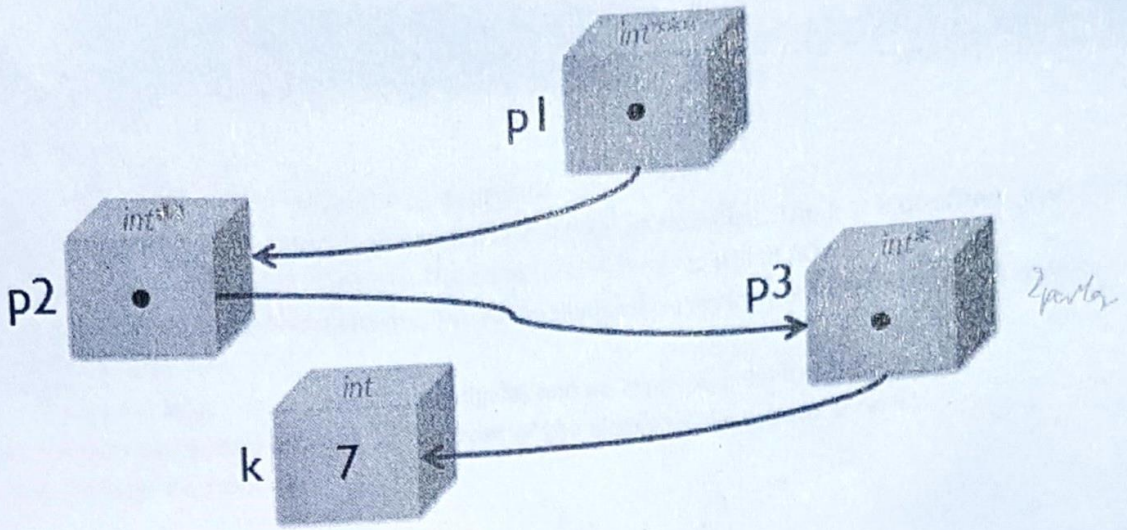
10

Cursor in new line after last character was printed

(not visible unless you print something else)

20

11. In the box below, realize the following memory configuration exactly using C variable declarations and pointer assignments. Then store 17, 18, and 19 into variable k using pointers p1, p2, and p3, respectively.



2 parts

```
int k = 7;
int** p3 = NULL;
*p3 = &k;
int** p2 = NULL;
**p2 = &p3;
int*** p1 = NULL;
***p1 = &p2;

***p1 = 17;
**p2 = 18;
*p3 = 19;
```

3
3
1

7