

APS 105H1 S 2010 Midterm

Duration — 90 minutes

Aids allowed: none

Student Number:

Last Name:

First Name:

Lecture Section: 1

Instructor: Daniel Zingaro

*Do **not** turn this page until you have received the signal to start.*

(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)

Good Luck!

This midterm consists of 8 questions on 14 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* If you use any space for rough work, indicate clearly what you want marked.

1: /15

2: / 6

3: / 4

4: / 7

5: / 7

6: / 5

7: / 6

8: / 5

TOTAL: /55

Question 1. [15 MARKS]**Part (a)** [2 MARKS]

Consider the following code fragment.

```
int i = 'c';
```

Are we allowed to assign a character constant to an `int` variable? Please first answer **yes** or **no**, and then explain your reasoning.

Part (b) [3 MARKS]

When we say `a = b`, `a` is assigned the value of `b`, and the expression evaluates to the value of `a` after the assignment.

Given this, carefully explain when the `printf` in the following code would execute.

```
if (a = b)
    printf ("The if-statement is executing.\n");
```

Part (c) [2 MARKS]

Fill in the following for-loop so that the `printf` statement is guaranteed never to execute. Your code must be legal C.

```
for (
    printf ("inside loop\n");
```

Part (d) [3 MARKS]

Briefly describe what is meant by short-circuit evaluation, and give a code fragment whose behavior depends on it.

Part (e) [2 MARKS]

What does this program fragment print?

```
int i = 3;
    int j = i++;
    int k = ++i;
    printf ("%d %d %d\n", i, j, k);
```

Part (f) [3 MARKS]

Any for-loop can be rewritten as an equivalent while-loop that does the same thing. Given this, explain why we would ever use a for-loop.

Question 2. [6 MARKS]**Part (a)** [3 MARKS]

Write a single C statement that declares a `bool` variable `b`, and assigns `true` to `b` if and only if variables `i` and `j` are not equal. Assume that `i` and `j` have already been declared and initialized.

Part (b) [3 MARKS]

Assume that we want to keep asking the user for an integer that is less than 0 or greater than 50, and store this number in `int` variable `num`. Give the while-loop condition that we would use.

`while (`

Question 3. [4 MARKS]

One kilogram equals 2.2 pounds, and one pound equals 16 ounces.

Write a complete program that asks for a `float` amount in kilograms, and prints that amount in pounds and in ounces. Use two decimal places in your output. Here is a sample program execution.

```
Enter an amount in kilograms: 4.0
Pounds: 8.80. Ounces: 140.80
```

```
int main(void) {
```

Question 4. [7 MARKS]

The speed limit on a highway is 100 km/h. If someone is caught speeding, they are given a fine that depends on their speed, as well as whether they have had a previous speeding ticket. The following table summarizes the fine that will be issued.

speed (km/h)	previous ticket?	fine amount
≤ 100	not applicable	\$0
101-120	no	\$50
> 120	no	\$80
> 100	yes	\$150

Write a program that asks for an integer traveling speed (in km/h), and a y or n response to whether they have a previous ticket. Your program should then output the amount of the fine (\$0, \$50, \$80, or \$150). Here is a sample program execution.

```
Traveling speed? 110
Do you have a previous ticket? y
You are fined $150!
```

```
int main(void) {
```

Question 5. [7 MARKS]

Write a function `bool allRepeated (int i)` that returns `true` if and only if positive integer `i` consists only of pairs of repeated digits. The following table gives several examples of what the function should return.

Function call	Returns
<code>allRepeated (8)</code>	<code>false</code>
<code>allRepeated (11)</code>	<code>true</code>
<code>allRepeated (2233)</code>	<code>true</code>
<code>allRepeated (22332)</code>	<code>false</code>
<code>allRepeated (222)</code>	<code>false</code>
<code>allRepeated (2323)</code>	<code>false</code>

```
bool allRepeated (int i) {
```

Question 6. [5 MARKS]

What is the output we get from running the following C program? Please show your intermediate work as you trace through the code.

```
#include <stdio.h>

int main (void) {
    int i, j, total = 0;
    for (i = 3; i <= 15; i = i + 2) {
        total = total + i;
        for (j = total; j <= total * 2; j++)
            i = total;
        printf ("%d\n", total);
    }
    return 0;
}
```

Question 7. [6 MARKS]

We would like to write a program that asks the user for a positive integer indicating how many dice to roll. If 0 is entered, the program stops; otherwise, the specified number of dice is rolled and the process is repeated. Here is a sample execution.

```
Roll how many dice (0 to stop)? 4
2 2 4 5
Roll how many dice (0 to stop)? 2
1 6
Roll how many dice (0 to stop)? 1
5
Roll how many dice (0 to stop)? 0
```

On the next page, you are given 12 fragments of code in random order. Your task is to use each of these fragments **at least once**, and add the proper opening and closing braces (`{ }`), to arrive at a program that implements the above description. Do **not** add new code; simply copy and rearrange the code from the fragments.

- `#include <stdio.h>`
 `#include <stdlib.h>`
 `#include <time.h>`
 `int main (void)`
- `do`
- `printf ("Roll how many dice (0 to stop)? ");`
 `scanf ("%d", &dice);`
- `srand(time(0));`
- `if (dice > 0)`
- `int counter = 0;`
- `int dice;`
- `printf ("%d ", rand () % 6 + 1);`
- `counter++;`
- `while (dice != 0);`
- `while (counter < dice);`
 `printf ("\n");`
- `return 0;`

Question 8. [5 MARKS]**Part (a)** [3 MARKS]

Write a function `void square (int val, int *where)` that sets the `int` pointed to by `where` to the square of `int val`. A code fragment containing a sample call of the function is as follows.

```
int a;  
square (3, &a); // a is now 9
```

```
void square (int val, int *where)
```

Part (b) [2 MARKS]

Why do we pass an `int` as the first parameter, rather than a pointer to that `int`?

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

Last Name: _____ **First Name:** _____