Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

GNG 1106C: Fundamentals of Engineering Computation

SAMPLE MIDTERM EXAM

Notes

1. NO calculators are allowed.
2. DO NOT un-staple, tear off, or otherwise separate pages!
3. This exam is closed book.
4. Questions are independent; i.e., the variables of one question do not carry over to the next.
5. Each multiple-choice question has only one correct answer (The most correct one). Read all choices before selecting an answer.
6. Enter the correct answer for multiple choice questions in the computer readable sheet.
7. Enter the answer of the programing question in this questionnaire.
8. Enter your name and student number in the fields below:

First Name: _____

Last Name: _____

Student Number: _____

Question 1

For the following questions, what is the output of the given program segment?

Q1.
```
int x = 130;

if (x >= 0 && x <= 100)
printf("Valid grade");
else
{
if (x < 0)
printf("Under range!!");
else
printf("Over range!!");
}
```

A. Valid grade
B. Under range!!
C. Over range!!
D. Under range!!Over range!!
E. None of the above.

Q2.
```
int i, j;

for (i = 2; i > 0; i--)
{
        for (j = 0; j < i; j++)
                printf("*");
                printf("%d ", i+j);
}
```

A. **4 **2
B. *2 *3 *1
C. **2 *3 1
D. **4 *2
E. None of the above.

Q3.
```
int a, b;
float z=0.0;
a = 4; b = 3;

z = a + b;
printf(“%f”, z);
```

A. 7
B. 7.000000
C. 4 + 3
D. 0.000000
E. None of the above.

Q4.
```
float x;
x = 0.35 == 3.5/10;
printf("%f",x);
```

A. 0.000000
B. 1.000000
C. 0.350000
D. 0.300000
E. None of the above.

Q5.
```
int num = 2;
int a = 5;
num = -a++ - num;
printf("%d,%d",num,a);
```

A. -7,6
B. -8,5
C. -8,6
D. -3,5
E. None of the above.

Question 2

Given the following program segment ; Answer question Q6.

```
#include <stdio.h>
int proc(int,int) ;

int main()
{
    int a,b,c;

    a = 35;
    b = 96;
    c = proc(++a,b);

    printf("%d:%d:%d",a,b,c);
}

int proc(int a, int b)
{
   int m = a;
   int n = b;

  while (m != n)
    if (m >  n)
        m = m – n;
    else
     n = n – m;
    return m;
}
```

Q6. What does the above program print?

A. 36:96:4
B. 36:96:12
C. 35:96:12
D. 36:96:4
E. None of the above..

Question 3

The program below is meant to implement the following: the program draws a random integer between 0 and 255 as a secret number; it then keeps prompting the user to guess the number until the user guesses it correctly. When the user's guess is wrong, the program prints a message "Your guess is low!" or "You guess is high!" to give the user a hint for his next guess. If the user guesses the secret number correctly, the program prints "You got it!" and the total number of guesses that user has made. In the following code, one function is left for you to implement. See the comments of the function to understand the requirements.

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

int letUserGuess (int);   /*prototype for the function letUserGuess. */

int main ()
{
   int numberOfGuesses ;
   int secretNumber ;
   secretNumber = rand () %256;
   numberOfGuesses = letUserGuess ( secretNumber );
   return 0;
}

/* This function keeps prompting the user to make his guesses until the user guesses it correctly.
When the user's guess is wrong, the program prints a message "Your guess is low!" or "You
guess is high!" to give the user a hint for his next guess. If the user guesses the secret number
correctly, the program prints "You got it!" and the total number of guesses that user has made.
The function returns the total number of guesses the user makes to get the secret number . */

int letUserGuess (int secretNumber )

/* fill in the code here */
```

The following table summarizes the order of precedence and the associativity of all the C operators that we have seen to date.

()	left to right	
- ! ++ -- (type)	right to left	**unary operators**
*** / %**	left to right	
+ -	left to right	
< <= > >=	left to right	
== !=	left to right	
&&	left to right	
||	left to right	
=, +=, -=, *=, /=, %=	right to left	