GNG1106C Midterm Solution, Fall 2014

Answers are in blue.

Question 1 (15 points) Check if the following code segments have errors.

```
float x;
x =100;
printf("%d", x);
```

No error. *%d* forces *printf* to treat *x* as an integer instead of a float, ignoring the fractional part of *x*. But in this specific case *x* does not have any fractional part, so *100* will be printed, which is correct.

```
int x;
scanf ("%d", x);
```

In the *scanf* statement, *&* is missing! It should be *scanf ("%d", &x);*

```
int x;
x=0;
while (1)
{
        x=x+2;
        printf ("x is %d\n", x);
        if (x >30)
                break ;
}
```

No errors. Compile the code and run, to see that it runs just fine.

Question 2 (21 points) What will the following code segments print respectively?

Hint: use a trace table, as shown in the class, to trace through the code and see the final answers.

```
int x=3, y=4, z=1;
if (z)
        printf ("x+y=%d\n", x+y);
else
        printf ("x-y=%d\n", x-y);
```

x+y=7

```
int x=0;
int i, j;
for (i=0; i <3; i++)
{
        x=x+2;
        for (j=1;j<x;j++)
```

```
                printf ("*");
        printf ("\n");
}
printf ("% d", x);
```

```
*
***
*****
```
6

```
int i=1;
while (i <6)
{
        switch (i)
        {
                case 1:
                        printf (" case 1\n");
                        i=i+1;
                        break ;
                case 2:
                        printf (" case 2\n");
                        i=i+2;
                        break ;
                case 3:
                        printf (" case 3\n");
                        break ;
                default :
                        printf (" default \n");
        }
        i++;
}
```

```
case 1
case 3
default
default
```

Question 3 (24 points) Give the output of each of the following programs.

```
# include <stdio.h>
int abc(int a, int b)
{
        int c;
        if (a<b)
                c=a+b;
        else
                c=a-b;
        return c;
```

```
}
int main ()
{
        int a, b;
        a=5;
        b=3;
        printf (" in main : %d", abc (a, b));
        return 0;
}
```

```
in main : 2
```

```
# include <stdio.h>
int abc(int a, int b)
{
        int c;
        printf (" entered abc : a is %d, b is %d\n", a, b);
        if (a<b)
                c=a+b;
        else
                c=a-b;
        a=c;
        b=c;
        printf (" exiting abc : a is %d, b is %d, c is %d\n", a, b, c);
        return c;
}
int main ()
{
        int a, b, c;
        a=5;
        b=3;
        c =10;
        c=abc(a, b);
        printf (" in main : %d\n", abc (abc (b, c), a));
        printf (" in main : a=%d, b=%d c=%d\n", a, b, c);
        return 0;
}
```

```
 entered abc : a is 5, b is 3
 exiting abc : a is 2, b is 2, c is 2
 entered abc : a is 3, b is 2
 exiting abc : a is 1, b is 1, c is 1
 entered abc : a is 1, b is 5
 exiting abc : a is 6, b is 6, c is 6
 in main : 6
 in main : a=5, b=3 c=2
```

```
#include <stdio.h>
int abc(int a, int b)
{
        static int c=8;
        printf (" entered abc : a is%d, b is %d\n", a, b);
        if (a<b)
                c=a+b;
        else
                c=a-b;
        a=c;
        b=c;
        printf (" exiting abc : a is %d, b is %d, c is%d\n", a, b, c);
        return c;
}
int main ()
{
        int a, b, c;
        a=5;
        b=3;
        c =10;
        printf (" in main : %d", abc (a, b));
        printf (" in main : %d", abc (abc (b, c), a));
        return 0;
}
```

```
entered abc : a is5, b is 3
exiting abc : a is 2, b is 2, c is2
in main : 2 entered abc : a is3, b is 10
exiting abc : a is 13, b is 13, c is13
entered abc : a is13, b is 5
exiting abc : a is 8, b is 8, c is8
in main : 8
```

Question 4 (40 points) 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 “Your 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 addition, based on the number of guesses the user made to get the secret number, the program also prints another message: if the number of guesses is 8, the program prints “Good job!"; if the number of guesses is less than 8, the program prints \You are lucky!"; if the number of guesses is more than 8, the program prints “You are ine_cient!". In the following code, two functions are left for you to implement. See the comments of the functions to understand the requirements.

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

int letUserGuess (int secretNumber )
/* This function keeps prompting the user to make his guesses
until the user guesses it correctly . It prints the hints and the message
```

```
"You got it !" as needed . The function returns
the total number of guesses the user makes to get the
secret number . */
{
        int numberOfTries=0;
          int userAnswer = -1;

          while (userAnswer!=secretNumber)
          {
            printf("\nPlease enter your guess: ");
            scanf("%d", &userAnswer);
            numberOfTries++;
            if (userAnswer==secretNumber)
              printf("\nYou got it!");
            else if (userAnswer<secretNumber)
              printf("\nYou guessed too low");
            else
              printf("\nYou guessed too high");
          }
          return numberOfTries;
}

void printMessage (int numberOfGuesses )
/* This function prints the " Good job !", "You are lucky !", or
"You are inefficient !" message based on the number of
guesses the user made . */
{
   if (numberOfGuesses==8)
      printf("\nGood job!");
   else if (numberOfGuesses<8)
      printf("\nYou are lucky!");
   else
      printf("\nYou are inefficient!");
}

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