Exercise 10 - Pointers

Objective

The major objective is to practice the declaration of pointers and the indirection and address operators, `*` and `&`.

Reference Material

This is based entirely on the *Pointers* chapter. This practical session is located in the following directory:

Windows Directory:	**c:\qacprogex\pointers**
Linux Directory:	**/home/user1/qacprg/POINTERS**

Overview

All questions are 'pencil and paper' exercises, and except for the optional question, which is a pointer to pointer example, they can be tackled in any order.

Practical Outline

1. Look at the following code fragments and answer the questions posed (assume these code fragments are within a `main`). To check your answers, open the Visual Studio Solution **pointers.sln**, and build / run the program.

a)
```
int m, n;
int *iptr;

m = 38;
iptr = &m;
n = *iptr;
```

What is the value of `n`?

b)
```
int m, n;
int *iptr;

n = 10;
iptr = &n;
n = 11;
m = *iptr;
```

What is the value of `m`?

c)
```
int *lptr;
int c, d;

c = 65;
lptr = &c;
d = *lptr + 1;
```

What is the value of d?

d)
```
int *lptr;
int i;

i = 65;
lptr = &i;
printf("%d", *lptr);
```

What will be displayed?

e)
```
int *lptr;
int i, j = 4;

lptr = &i;
i = j;
printf("%d", *lptr);
```

What will be displayed?

f)
```
int num_days, i = 4;
int *lptr = &num_days;

printf("%d", *lptr);
```

What will be displayed?

g)
```
float f = 4, fred = 37;
int *lptr;

f = fred;
printf("%d", *lptr);
```

What will be displayed?

h)
```
int i = 9, j = 10;
int *lptr = &i;

*lptr = i;
j = i;
printf("%d %d %d", *lptr, i, j);
```

What will be displayed?

i)
```
int i, j;
int  *p1 = &i, *p2 = &j;

*p1 = 8;
i = 7;
*p2 = *p1;
```

What is the value of `i` and `j`?

j)
```
float zero = 1.0, one;
float *fp1 = &zero, *fp0 = &one;

fp1 = fp0;
*fp1 = 0;
*fp0 = 1;
```

What is the value in `zero` and `one`?

k)
```
char d, ch = 'q', grade = 'b';
char *cp, *pp;

cp = &grade;
grade = 'l';
d = *cp;
pp = cp;
*pp = grade;
printf("%c %c %c %c %c", d, ch, grade, *cp, *pp);
```

What will be displayed?

l)
```
float f, f1 = 4.0, f2 = 1.5;
float *fp1 = &f1, *fp2 = &f2;

f = *fp1 * *fp2;
```

What is the value in `f`?

m)
```
long lval1 = 3, lval2 = 2, *lptr;

lptr = &lval1;
*lptr = lval2++ * *lptr;
```

What is the value in `lval1` and `lval2`?

Optional

n)
```
long lval1, lval2 = 4, *lp = &lval2, **lpp = &lp;

lp = &lval1;
*lp = 2;
**lpp = 3;
```

What is the value of `lval1` and `lval2`?