Exercise 4 – Declarations

Objective

The major objective is to practice the reading and writing of complex declarations.

Reference Material

This practical session is based entirely on material in the Declarations chapter. The session is located in the following directory:

	Microsoft Windows	*Linux*
Directory:	***c:\qacadv\decls***	***~/qacadv/decls***

Overview

All questions are 'pencil and paper' exercises. Online solutions are provided, so you can check your answers.

Practical Outline

1. Describe the types of the following variables:

```
int         *a;
int         *b[5];
int         (*c)[5];
float       d(void);
float       *e(void);
float       (*f)(void);
float       *(*g)(void);
float       (**h)(void);
char        (*i(void))[3];
char        *(*j(void))[3];
```

   To check your answers for this (and the following) questions:

   On Microsoft Windows launch Visual Studio and open **c:\qacadv\decls\decls.sln** and build the program.

   On Linux change directory to **~/qacadv/decls** (the program **decls** is already built).

   Run the **decls** program to see the correct answers for this question.

2. Given the line of code:

```
x = (int (*)(int*))y;
```

What type would `x` have to be (assuming there are no compiler warnings)?

Given the two lines:

```
typedef void    (*PFVV)(void);
PFVV            z;
```

what type would the variable `z` have?

3. Declarations for the variables described below :

k: an array of 6 pointers to char
l: a pointer to an array of 6 pointers to char
m: an array of 5 pointers to arrays of 6 pointers to char
n: a function, taking no parameters, returning a pointer to char
o: a function, taking no parameters, returning a pointer to an array of 6 pointer to char
p: a function, taking no parameters, returning a pointer to a function, taking no parameters, returning an int
q: a function, taking no parameters, returning a pointer to a function, taking no parameters, returning a pointer to an int
r: a function, taking no parameters, returning a pointer to a function,taking no parameters, returning a pointer to an array of 6 pointer to char

4. Create a `typedef` for the variable `l` declared above so that it may be declared as follows:

```
PA6C          l;
```

Use this typedef of `PA6C` to simplify the declarations of `m`, `o` and `r`

Even using this typedef, the declaration of `r` is still difficult to read. Create another intermediate typedef to make the declaration easier.