Part A: Pointers

1) a) Write a line of code using the “typedef” command

int *p Pointer P

b)Explain why the “typedef” command might be useful when dealing with pointers

typedef will help with no confusion when declaring pointer and also when creating alias type definition if we need to send a pointer as a parameter to a function.

2) Assume the following code has been run:

```
  int *p1, *p2, v1, v2;
2 p1 = &v1;
3 *p1 = 42;
```

a) Explain what happen in lines 2 and 3

at line 2 assign pointer p1 to point at the address of v1 and get the value of v1
at line 3 assign value 42 into pointer p1

b) What is the difference between ‘p2=p1;’ and ‘*p2=*p1;’ assuming they follow the above code?

p2=p1 : pointer p2 pointing at the same address as p1.
*p1=*p2 : assign value of pointer p1 to p2.

3) What is a dangling pointer?

the pointer that doesn't point anywhere or the pointer that point to the memory that has been deleted.

4) a) Suggest a situation where you might need a “Destructor”

when dealing with dynamic array

b) If the class is called ‘person’, what should your destructor function be named?

~person

c) Given a destructor, why would you need a ‘copy constructor’?

When the destructor called it will delete the object that passed out of the scope therefore when we use that object again, it will be undefined. So we need to use Copyconstructor to create an object by initializing an object of the same class that have been created previously.