Question 1: Match the columns

Column A

Memory address of integer variable a

Memory address of a variable
pointed to by pointer p   **Column B**

Value stored at the address pointed
to by pointer p   p

Comparison to determine whether   a

Pointers p and q point to the same   *p

variable   *a

Comparison to determine whether   &p

Pointers p and q point to variables   &a

that have the same values   p==q

Comparison to determine whether   &p==&q

Pointers p and q have the same   *p==*q

contents

Question 2: Assume the variable declarations:

```
int Foo = 0;
int *ptr = &Foo;
```

Which of the following statements will change the value of Foo to 1? Write all possibilities

```
1) ptr++;                      3) (*Foo)++;
2) Foo++;                      4) (*ptr)++;
```

Question 3: Write a function `func()` that takes as argument the size of an int array and allocates an array of this size dynamically. The function sets all elements of the array to 0 and returns a pointer to the array. Where should this array be de-allocated (in main or in the function `func()`) ?

Question 4: Consider the following allocations.

```
string *p1 = new string("one"); // Line 1
string *p2 = new string("two"); // 2
p2 = new string("three"); // 3
p2 = p1; // 4
delete p2;    // 5
```

Indicate, on which lines, a **memory leak** occurs or a **dangling pointer** is created.