1. Perl, Python, Ruby
2. Because this is a fundamental need for many computer application so allowing and programing for such feature is necessary. Its also easier because strings have strict definitions. They are always one dimensional have exactly one byte elements and never contain any references to any other objects.
3. Depending on the size of each object in the bit-vector it can consume a lot of memory. For example a 64 bit set of integers would consume more memory then that is available on any system. Usually bit vectors are used for 7 or 8 bit units like characters. The advantage to bit-vectors is they help perform logic functions on the items in the set extremely fast.
4. The main difference that seems to exist is that using pointers requires the program to be aware of all the references they create. No deleting a point after it is no longer in use could cause a memory leak or dangling pointer. The languages that use recursive types generally have some sort of garbage collection within the language itself which makes it easier on the programmer. This comes at the cost of the runtime processing power.
5. Clu, Cedar, Modula-3, Java and C# all just use automated garbage collection to handle dangling pointers and dereferenced values.

Pascel uses a dispose function: dispose(my\_ptr);

C uses the free function: free(my\_ptr);

And C++ uses the delete keyword: delete my\_ptr;