Question 2 Part C answer:

Reasons for Superiority of C++ exception handling over Setjmp () – Longjmp () exception handling:

1. Exception handling in C++ via try-catch mechanism induces of semantically much better codes, which provide easier understandability and ease of access and use.

2. If the function in which setjmp() was called returns normally and longjmp() with the corresponding jmp_buf object is called then longjmp() restores the stack pointer of the stack frame which is invalidated when the function returns normally, where stack pointer would point to a non-existent and potentially overwritten/corrupted stack frame.

3. A problem with the use of setjmp/longjmp is that cleanup (such as closing file descriptors, flushing buffers, freeing heap-allocated memory, etc.) is not conducted automatically.
   Which is handled in exceptions in C++ called Stack Unwinding.

4. In contrast to DWARF exception handling, which encodes exception regions and frame information in out-of-line tables, SJLJ (setjmp () – longjmp ()) exception handling builds and removes the unwind frame context at runtime. This results in faster exception handling at the expense of slower execution when no exceptions are thrown. As exceptions are, by their nature, intended for uncommon code paths, DWARF exception handling is generally preferred to SJLJ.

-Siddharth Rakesh 11CS30036