CS520 Spring 2015 Final Exam

- **This exam has a total of 100 points. There are twenty questions, which are all worth 5 points each.**
- **Answer the questions on separate paper.**
- **For partial credit show your work.**
- **The exam is closed book and notes.**
- **Please keep all electronic devices turned off and out of reach.**
- **When turning in your exam, please fold your papers in half, left-to-right, and write your name on the outside.**
- **I will return your exam to your Kingsbury mailbox unless you tell me not to.**

1. (5 points) Write a C function called `isInfinity` that will test a 32-bit IEEE single-precision floating-point value and will return 1 if it is either positive or minus infinity, and return 0 otherwise. The function should only use integer operations. The function should take one `unsigned int` argument. The argument will contain 32 bits that should be interpreted as a floating-point value.

2. (5 points) In our implementation of a threads library, what role did the *thread_start* function play? Explain why it was useful even though it was never called and it never returned.

3. (5 points) Decode the following UTF-8 sequence (shown as hexadecimal) to a Unicode character: `0xD0 0xA1`. Show the Unicode character as hexadecimal.

4. (5 points) Write an Intel 64 assembly language function called `addIndirect` that will take two arguments: a 64-bit address as the first argument and a 64-bit integer value as the second argument. The function should add the 64-bit integer value given by the second argument and the 64-bit integer value stored at the address given by the first argument. The function should return the result of the addition.

5. (5 points) Write a C function called `isLittleEndian` that will return 1 if the machine executing the function is Little Endian and 0 otherwise.

6. (5 points) In an implementation of a threads package, how would the `condWait` function be implemented? Provide pseudo code for the function. Also explain what fields are stored in the memory allocated to represent a condition variable. You may assume that the threads package will run on only a single processor.

7. (5 points) Add together the following two IEEE single-precision floating-point values: 40000000 and 40000003. Show the IEEE single-precision floating-point result in hexadecimal.

8. (5 points) Decode this RISC-V instruction (shown in hex): `FEFC8193`.

9. (5 points) In an implementation of an exception mechanism for C programs, explain exactly what the `throwException` function would do.

10. (5 points) Explain the differences between a direct-mapped cache, a set-associative cache, and a fully associative memory cache.

11. (5 points) What is the purpose of the mark phase of a mark-and-sweep garbage collector?

12. (5 points) Von Neumann's design for the IAS computer included an instruction to do address modification. What did this instruction do exactly?

13. (5 points) Consider how the following two C loops would be accessed by a direct-mapped memory system with 8 sets and a block size of 8 ints.

```
for (i = 0; i < 256; i++)
  a[i] = i;
sum = 0;
for (i = 0; i < 256; i++)
  sum += a[i];
```

Assume only the array is in the cache and the first word of the array is at address zero. If the cache is initially empty, how many cache hits would there be for writes and how many cache hits would there be for reads?

14. (5 points) Show how -17 (base 10) would be represented in memory as a 16-bit 2's complement integer. Show your answer in hexadecimal and show all the hex digits, even if they are zero. Clearly label the order in which the bytes would lie in memory. Assume the machine is Little Endian.

15. (5 points) Write a C function prototype declaration for a function called `thread_create` that takes two parameters and returns a `void*`. The first parameter is a pointer to a function that takes a single parameter of type `void*` and returns nothing (has a `void` return type). The second parameter is of type `void*`. (In other words, provide the prototype declaration for the `thread_create` function from Program 5.)

16. (5 points) Convert the UTF-16 sequence (shown in hexadecimal) `0xD859 0xDE66`. to UTF-32. Show your answer in hexadecimal. Be sure to clearly indicate how many UTF-32 characters are in the result.

17. (5 points) Describe in POSIX threads pseudo-code how to implement the producer-consumer pattern when there are multiple producers and multiple consumers. Assume the producers are placing values in a fixed-length array and the consumers are removing values from the array. You do not need to worry about termination.

18. (5 points) In an object file, what is an insymbol? What is an outsymbol?

19. (5 points) How does a virtual memory system try to exploit spatial locality? How does it try to exploit temporal locality?

20. (5 points) Add together the following two 8-bit 2's complement integers (shown in hexadecimal): `0x70` and `0x1B`. Show the 8-bit 2's complement answer in hexadecimal and show all the hex digits, even if they are zero. Does the answer overflow? Please indicate either yes or no. Also check your answer using decimal: What decimal values do the two input values represent? What is the answer in decimal?