Show all your work in addition to your answers. Closed books/notes and no electronic aids except for a non-programmable calculator. Use backside of this page if necessary. Each question is worth 10 points.

1. What is the limit on input frequency to the PLL multiplier in the PIC32MX clock module?
4 MHz. The PLL circuitry can only handle input frequencies ≤ 4 MHz. See lecture slide 7.

2. What are the two major components in the power consumption of a CMOS device such as PIC32MX?
Which of these components depends on the device frequency?
Total power = static power + dynamic power. Dynamic power depends on frequency.

3. The Secondary Oscillator Clock Chain (SOSC) works with the OSCCON SFR to MUL and DIV to set the system and peripheral clock rates. Is this true? Justify in sufficient details if your answer is False or No.
False. It is the POSC that works with the OSCCON SFR. Also, only the POSC has the ability to MUL and DIV to generate system and peripheral clock rates, not SOSC.

4. String literals such as “PIC32MX” or “cached variables” are placed in the .data section of the program memory. Is this true? Justify in sufficient details if your answer is False or No.
No. String literals are constants and must be placed in .rodata section (read only), not .data section.

5. In the PIC32MX memory map, .vector_x can contain up to 64 sections, each of which relates to an ISR used in the PIC32MX program. Justify in sufficient details if your answer is False or No.
Yes. See p. 125 of Di Jasio.

6. To use dynamic (defined when program is running) data structures, we have to use the run-time stack. Is this true? Justify your answer in sufficient details if your answer is False or No.
No. Heap, not stack, has to be used for dynamic data structures.

7. What should be done about dynamic data structures before a program exits? What will happen if this is not done?
These data structures must be released/freed before exiting a program. If not, the next programs will have less and less memory for their dynamic data structures, a.k.a. memory leak.

8. What does the C statement `ptr = malloc(sizeof(int)*15);` do? Explain in sufficient details.
This statement dynamically allocates a block of 4*15=60 bytes on the heap and returns the starting address of the block of memory to `ptr`. See p. 128 of Di Jasio.

9. For `const char myArray[10]`, it will be initialized after the execution of `crt0` (i.e. prologue) code. Justify your answer in sufficient details if your answer is False or No.
No. Constant data are allocated and initialized at build time, i.e. before the `crt0` code is executed. Variable data, on the other hand, are initialized at run time, i.e. after the `crt0` code is executed. See lecture slides 2-5 for details.
10. For embedded processors, the data type char is often used to store memory addresses rather than characters as the data type is originally designed for. Why is this?

The data type char is 1-byte wide and the type is compatible with the int type. However, char only takes 1/4 of the space of int which is 4-byte wide. Thus, for small processors with small memory sizes, such as embedded processors, char can be used to sufficiently store memory addresses and save on overall memory usage.