

1. The ISA instructions that a computer executes **consist of** a(n) _____, and optionally, a(n) _____.

2. **What type of opcode** has a different number of bits in it depending on how many operands the instruction has?

3. Does the **Pentium-4** have **fixed** or **variable-length** instructions?

4. Does the **UltraSPARC** have **fixed** or **variable-length** instructions?

5. **How long** (how many bits) is an instruction in a computer system that has **64 registers** if the opcode is **12 bits long** and the instruction specifies **3 registers**?

6. Write the name of the addressing mode beside each of the descriptions below:

	The data is located in memory at the address held in a register specified by the instruction.
	The data is located in the register specified by the instruction.
	The data is located in memory at the address specified in the instruction.
	The data is located in memory at the address which is computed by adding the contents of a register specified by the instruction to an offset specified in the instruction.
	The data is specified in the instruction.

7. What type of addressing uses PUSH and POP instructions:

8. Compute the value of the following postfix expression:

26 15 10 + 5 4 - + / 8 *

9. What are the **names** of these types of **branch addressing** :

- a) The value in the instruction is **put into** the Program Counter
- b) The value in the instruction is **added to** the Program Counter

10. How does a **CALL** instruction **differ** from a **GOTO** instruction?

11. **Where** is the return address (used by the **RETURN** instruction) stored in most Instruction Set Architectures?

12. **What is wrong** with the following sequence of instructions:

	MOV	CX, FILE_SIZE	!Put file size into the CX register
	MOV	SI,BUFFER	!Point SI to the start of the file buffer
A10:	MOV	AL,[SI]	!Put a byte from file into the AL register
	CALL	PROCESS_BYTE	!Process the byte in the AL register
	ADD	SI,1	!Point SI to the next byte
	SUB	CX,1	!Subtract 1 from no. of bytes remaining
	JNZ	A10	!Loop for next byte if not end

13. **What is it called** when **every byte** of data read from or written to an I/O device is transferred by **instructions executed by the CPU**?
14. One way to tell when an output device is ready to accept more data is to “**poll**” the device (request and check it’s status **over and over again** until it’s ready).
What’s a more efficient alternative?
15. **What event** causes an Interrupt Service Routine to start executing?
16. What is the difference between a normal procedure and a **recursive** procedure?
17. Show the first 10 numbers that would be printed by the following subroutine when it’s called with parameters of 0 and 1:
- ```
int fib(int Param1, int Param2)
{
 print(Param1); // Prints value of Param1 followed by a space
 fib(Param2, Param1+Param2)
}
```
18. **Why** would the above subroutine **fail with an error**?

