



### Objective:

- The purpose of this quiz is to focus on the very basic fundamental concepts learned so far in previous lectures.

### Question:

In computing, **endian** and **endianness** in the most common cases refer to how bytes are ordered within a data item, and *endianness* is then the same as **byte order**. A **big-endian** machine stores the most significant byte first—at the lowest byte address—while a **little-endian** machine stores the least significant byte first.

Endian	First byte (lowest address)	Middle bytes	Last byte (highest address)	Decimal 100000000 (hexadecimal 05F5E100)										
big	most significant	...	least significant	<table><tr><th>Address</th><th>Value</th></tr><tr><td>n</td><td>05</td></tr><tr><td>n+1</td><td>F5</td></tr><tr><td>n+2</td><td>E1</td></tr><tr><td>n+3</td><td>00</td></tr></table>	Address	Value	n	05	n+1	F5	n+2	E1	n+3	00
				Address	Value									
				n	05									
				n+1	F5									
				n+2	E1									
n+3	00													
little	least significant	...	most significant	<table><tr><th>Address</th><th>Value</th></tr><tr><td>n</td><td>00</td></tr><tr><td>n+1</td><td>E1</td></tr><tr><td>n+2</td><td>F5</td></tr><tr><td>n+3</td><td>05</td></tr></table>	Address	Value	n	00	n+1	E1	n+2	F5	n+3	05
				Address	Value									
				n	00									
				n+1	E1									
				n+2	F5									
n+3	05													

```
void main()
{
    int a=65;
    cout<<"a Address : "<<&a; //Assumes address of a is 100

    cout<<endl<<(void*)((char*)&a+0)<<"\t"<<*((char*)&a+0);
    cout<<endl<<(void*)((char*)&a+1)<<"\t"<<*((char*)&a+1);
    cout<<endl<<(void*)((char*)&a+2)<<"\t"<<*((char*)&a+2);
    cout<<endl<<(void*)((char*)&a+3)<<"\t"<<*((char*)&a+3);

    if ((void*)((char*)&a+0)<(void*)((char*)&a+1))
        cout<<"\nMachine is Little Endian";
}
```

**a Address : 100**  
**100     A**  
**101     0**  
**102     0**  
**103     0**  
**Machine is Little Endian**