**Research Questions**

1. The Intel 8085 microprocessor was a first generation processor that was used in many early game systems and personal computers. Google “8085 microprocessor architecture” to answer these questions.

a. Year Introduced

**1976**

b. Size of data bus (in bits)

**8-bits**

c. Largest data number (in binary and decimal)

**65,535 Decimal and 16 Binary**

d. Size of address bus (in bits)

**8-bits**

e. Largest memory address (in binary and decimal)

**65,535 Decimal and 16 Binary**

2. The Intel 8086 microprocessor was the processor used in the first IBM PCs running the DOS operating system. Google “8086 microprocessor architecture” to answer these questions.

a. Year Introduced

**1979**

b. Size of data bus (in bits)

**16-bits**

c. Largest data number (in decimal)

**1,048,576**

d. Size of address bus (in bits)

**20-bit**

e. Largest memory address (in decimal)

**1,048,576**

3. The Intel 80286 microprocessor a common processor used in IBM PCs running the Windows operating system. Google “80286 microprocessor architecture” to answer these questions.

a. Year Introduced

**1982**

b. Size of data bus (in bits)

**16-bits**

c. Largest data number (in decimal)

**65,535**

d. Size of address bus (in bits)

**24-bits**

e. Largest memory address (in decimal)

**65,535**

4. The modern PCs run either a 32 bit or 64 bit Windows operating system. Google “32 vs 64 bit” to answer these questions.

a. How do these systems differ in data capacity? (explain using bits)

**64-bit processors are more capable than a 32-bit processor because it can handle more data at once.**

b. How do these systems differ in memory capacity? (explain using bits)

**A 64-bit processor of storing more computational values, including memory addresses, which means it is able to access over four billion times the physical memory of a 32-bit processor.**

c. How do these systems differ in hardware requirements?

**32-bit processors are perfectly capable of handling a limited amount of RAM (in Windows, 4GB or less) and a 64-bit processor is capable of utilizing much more. In the case of Microsoft Windows, the basic versions of the operating systems put software limitations on the amount of RAM that can be used by applications, but even in the ultimate and professional version of the OS, 4GB is the maximum usable memory the 32-bit version can handle. The latest versions of a 64-bit operating system can increase the capabilities of a processor drastically.**

5. Research and explain how negative (-) numbers are represented using bits and how they are stored in computer memory.

**Negative numbers in any base are represented by prefixing them with a minus (-) sign. In computer hardware, numbers are represented only as sequences as bits, without extra symbols. Computers can only store information in bits, which can only have the values of zero or one.**

6. Research and explain how floating point (decimal) numbers are represented using bits and how they are stored in computer memory.

**Eight digits are used to represent a floating point number : two for the exponent and six for the mantissa. The sign of the mantissa will be represented as + or -, but in the computer it is represented by a bit: 1 means negative, 0 means positive. This representation makes it easy to compare numbers. When looking at the mantissa (the value between 1.0 and (almost) 2.0), one sees that all possible values start with a "1" (both in the decimal and binary representation). This means that it's no point in storing it. The rest of the binary digits are stored in an integer field, in the 32-bit case this field is 23 bits.**

SAMPLE PROGRAM for BINARY:

number = input ("Enter a 4 digit decimal")

number: ")"

index = 0

for x in number:

index += 1

print ("Digit ", index, "is : ", bin(int(x)))

SAMPLE PROGRAM for OCTAL:

number = input ("Enter a 4 digit decimal")

number: ")"

index = 0

for x in number:

index += 1

print ("Digit ", index, "is : ", oct(int(x)))

SAMPLE PROGRAM for HEX:

number = input ("Enter a 4 digit decimal")

number: ")"

index = 0

for x in number:

index += 1

print ("Digit ", index, "is : ", hex(int(x)))