1. What is the architecture of a processor?

A. the style and colors for the case and monitor.

B. the type of operating system and software that the computer runs.

C. the shape of the chip and the layout of its connector pins.

**D. a description of the basic components of a processor chip and of its basic operations.**

2. What language allows the programmer to program the processor at the architectural level?

**A. assembly language**

B. Java

C. FORTRAN

D. C

3. What is the machine cycle?

A. One machine cycle consists of all the steps taken in executing a program.

B. A machine cycle is how a cold computer is booted into running an operating system.

**C. A machine cycle is the process through which one machine instruction is executed.**

D. A machine cycle is the steps through which a byte is fetched from memory.

4. What are the three steps in the machine cycle?

**A. fetch, increment, execute**

B. increment, fetch, execute

C. load, compile, run

D. wash, rinse, spin dry

5. What is a machine instruction?

A. a pattern of bits that corresponds to one unit of data.

B. a signal sent across the system bus that controls the operation of the computer.

**C. a pattern of bits that asks for one fundamental operation of the processor**.

D. a signal sent by the system clock that starts one machine cycle.

6. What part of the processor indicates which machine instruction is next in line for execution?

A. The address bus.

B. The memory address register.

**C. The program counter.**

D. The system clock.

7. What is a register?

A. a part of the processor that performs an operation.

B. a part of the processor that keeps a log of operations.

C. the part of the operating system that oversees what programs are selected for execution.

**D. a part of the processor that holds a bit pattern.**

8. Do all processor chips use the same machine language?

**A. No. Machine language is an engineering decision that is unique to each processor family.**

B. No. Each individual processor chip has its own unique machine language.

C. Yes. Machine language is a fundamental characteristic of all processors.

D. Yes. An International Committee designed the machine language that all processors now use (after the year 2000).

9. Assemblers and compilers usually translate a source program into machine instructions contained in what type of file?

A. include file

B. binary file

**C. object module**

D. hidden file

10. What type of program uses software to imitate the hardware operation of a particular type of processor?

A. imitator

B. assembler

C. translator

**D. emulator**

1. What does the word binary mean?

A. Binary means "containing a computer."

**B. Binary means "having only two states."**

C. Binary means "having a discrete number of values."

D. Binary means "using electronics to do arithmetic."

2. What is a bit?

**A. A bit is a single binary value.**

B. A bit is a collection of several bytes.

C. A bit is a single character stored in main memory.

D. A bit is a small unit of computer time.

3. Which of the following is NOT an advantage of building computers out of binary devices?

A. Binary devices are simple and easy to build.

B. Binary signals are unambiguous.

**C. Binary devices are much faster than decimal devices.**

D. Patterns of bits can be used to represent anything symbolic.

4. What is true of an analog signal?

A. An analog signal has a discrete number of states.

B. An analog signal is the only way that music can be recorded.

C. An analog signal can never be converted into a binary signal.

**D. An analog signal is usually continuously changing in value.**

5. If an analog signal picks up some noise, has information been lost?

A. No — electronics can just ignore the noise.

B. No — information has been added to the signal.

C. Maybe — it depends on how loud the noise is.

**D. Yes — the noise hides the exact values of the original signal.**

6. If a binary signal picks up some noise, has information been lost?

**A. No — the exact value of the bits can be determined, as long as the noise is not too great.**

B. No — binary signals can't pick up any noise.

C. Yes — the exact value of the bits cannot be determined.

D. Yes — the signal will have extra bits in it because of the noise.

7. Why does a computer have a clock?

**A. The state of binary signals is measured only at specific instants in time.**

B. A clock is needed to check how fast signals are changing.

C. A clock is needed to check that voltage levels are correct.

D. A clock is used only with application programs that need to know the current time.

8. Can Japanese writing be represented in a computer?

A. No — only English and English-like languages can be represented.

B. No — only languages with an alphabet can be represented.

C. Yes — but a special processor chip is needed.

**D. Yes — since it is symbolic, and anything symbolic can be represented.**

9. Can English writing be represented with analog signals?

A. No—only binary signals can represent symbolic data.

B. No — it is not symbolic.

C. No — analog signals don't represent anything.

**D. Yes — just read out loud into a microphone.**

10. Why is it important that unlimited perfect copies can be made of data represented in binary?

A. Transmitting data over the Internet involves making many copies of the data.

B. Application programs such as wordprocessors and computer games must be perfect copies of the original in order to run.

C. Because data are copied back and forth between parts of a computer system many times per second.

**D. All of the above.**

1. How many patterns can be formed with a single bit?

A. 1

**B. 2**

C. 4

D. 8

2. How many patterns can be formed with three bits?

A. 1

B. 3

**C. 8**

D. 12

3. How many patterns can be formed from N bits?

A. N

B. N2

**C. 2N**

D. 2\*N

4. What is 24 \* 23?

A. 82

B. 26

**C. 27**

D. 212

5. How many bits are in a nibble?

A. 2

**B. 4**

C. 8

D. 16

6. What is the name for the bit pattern 0000 0010 ?

A. 0x0002

**B. 0x02**

C. 0xA0

D. 0x201

7. What is the name for the bit pattern 101010?

A. 0xA2

B. 0x222

C. This pattern can not be named.

**D. 0x2A**

8. Say that you read that a particular computer "uses hexadecimal". What is being described?

A. ...the fundamental hardware of the processor chip.

**B. ...a convention used in written documentation of the computer.**

C. ...the type of arithmetic carried out by the computer.

D. ...the type of signals used on the computer's bus.

9. What is the name for the convention where bits are named in groups of three?

A. Trinary

B. Sextuple

C. Tridecimal

**D. Octal**

10. How many bytes are in a kilobyte?

A. 8

B. 1000

**C. 1024**

D. 4068

1. What is a the name for a group of parallel conductors on the main circuit board over which data and signals flow?

A. PCB

**B. bus**

C. system cable

D. device controller

2. In which of the following lists are devices arranged in order of slowest to fastest for data movement into and out of the processor?

A. main memory, hard drive, floppy disk

B. floppy disk, main memory, hard drive

**C. floppy disk, hard drive, main memory**

D. hard drive, main memory, floppy disk

3. What is the name for an electronic module that responds to requests from the central processor, by sending device-specific control signals to an I/O device?

A. bus handler

B. I/O module

C. virtual device

**D. device controller**

4. What is the main difference between the section of memory that holds instructions and the section of memory that holds data?

A. Data memory is arranged into bytes with addresses; instruction memory holds words without addresses.

**B. There is no difference. All memory does is hold bit patterns. It is up to the rest of the computer system to determine what those patterns mean.**

C. Data memory is connected to the data bus. Instruction memory is connected to the instruction bus.

D. Data uses virtual memory; instructions use physical memory.

5. In most modern processors, what is the smallest addressable unit of memory?

A. bit

**B. byte**

C. 32 bits

D. 64 bits

6. How many memory addresses are there with a 32-bit bus?

**A. 232**

B. 32

C. 322

D. This depends on how much memory has been installed in the computer.

7. How many bits are in the addresses of the MIPS 32 chip, the processor that is the subject of this course?

A. 16

B. 24

**C. 32**

D. 64

8. On a modern computer, the processor "sees" the full address space even though there is less installed memory than that. What is this full address space called?

**A. virtual memory**

B. apparent memory

C. RAM

D. physical memory

9. How is the illusion of a full address space maintained on a modern computer?

A. Addresses on the system bus are divided between disk storage and RAM memory.

**B. The operating system, with the help of special electronics, uses both physical memory and disk storage to implement the address space.**

C. Most programs use far less than the full address space, so they don't notice that only part of it is there.

D. The processor divides time into very small slices and changes the contents of memory after each slice.

10. What is cache memory?

A. The part of the hard disk that lists its contents.

B. The part of the hard disk used for virtual memory.

C. Read-only memory that holds basic I/O instructions.

**D. Memory inside the processor chip that is used to speed up main memory access.**

1. How many bits are in a byte on most computers?

A. 2

B. 4

C. 6

**D. 8**

2. About how much data does one byte correspond to?

**A. One character.**

B. One floating point number.

C. One machine instruction.

D. One signal.

3. What is the NUL byte?

**A. a byte that contains all zeros.**

B. a byte that contains all ones.

C. a byte that contains no bits.

D. a byte that contains any illegal pattern.

4. What is a control character?

A. A byte pattern that corresponds to any non-alphabetic and non-digit character.

B. The signal that shifts the carriage from lower case to upper case.

**C. A byte pattern that originally was intended to ask for a mechanical action of an output device.**

D. Any character that takes up more than one byte.

5. What is a teletype machine?

A. An early model of personal computer.

B. An early type of printer that punched holes instead of using ink.

**C. A mechanical device used to send and receive printed text across telegraph lines.**

D. An early adding machine that used mechnical parts rather than electronics.

6. What is the name of the convention that says how bit patterns are used to represent characters.?

A. DOS

**B. ASCII**

C. HTML

D. AIISC

7. If characters are arranged in numeric order, what is the first printable character?

A. A

B. 0

C. @

**D. space**

8. What type of file that contains only bytes that correspond to printable characters and a small number of control characters?

**A. text file**

B. type file

C. control file

D. word file

9. What do people sometimes call a file that contains bytes that can potentially hold any bit pattern?

A. text file

B. data file

**C. binary file**

D. ASCII file

10. How many patterns can be made out of eight bits?

A. 8

B. 128

**C. 256**

D. 1024

1. Which of the following could represent the number eight?

A. 8

B. //// ////

C. iix

**D. They all could represent the number.**

2. The number thirteen is a prime number, a number that has no divisors other than itself and one. Which of the following represent a prime number?

A. 13

B. XIII

C. 11012

**D. All of them. The properties of a number do not depend on how it is represented**.

3. What are some of the advantages of positional notation?

**A. Compact, easy to compute with**.

B. Uses familiar characters, always uses base ten.

C. Does not need a symbol for zero, easy to learn.

D. Numbers of any size can be represented, more accurate than other methods.

4. What is the value of B0, where B is an integer?

A. 0

**B. 1**

C. 2

D. B

5. What are the "digits" ordinarily used for base five?

**A. 0 1 2 3 4**

B. 1 2 3 4 5

C. 0 1 2 3 4 5

D. a b c d e

6. Here is a number represented in base three. The expression on the right of = is in base ten. What should fill the blanks?

2013 = 2 × 3--- + 0 × 3--- + 1 × 3---

**A. 2 1 0**

B. 3 2 1

C. 1 2 3

D. 0 1 2

7. Convert the following to base ten: 2013

A. 5

B. 8

**C. 19**

D. 98

E. 105

8. Change the representation of 2034 from base four to base ten.

A. 510

B. 1110

C. 1910

**D. 3510**

E. 6710

9. What are the "digits" of base two number representations usually called?

**A. bits**

B. bytes

C. quarks

D. dots

10. Change the representation of 102 from base two to base ten.

A. 0

B. 1

**C. 2**

D. 3

E. 4

F. 5

11. Change the representation of 1012 from base two to base ten.

A. 0

B. 1

C. 2

D. 3

E. 4

**F. 5**

12. Change the representation of 10112 from base two to base ten.

A. 310

B. 810

**C. 1110**

D. 1410

E. 1910

F. 3210

13. Change the representation of 248 from base eight to base ten.

A. 710

B. 1210

C. 1710

**D. 2010**

E. 2710

F. 3210

14. Change the representation of 0101002 from base two to base ten.

A. 1710

**B. 2010**

C. 2110

D. 2610

E. 2810

F. 3210

15. Change the representation of 248 from base eight to base two (hint: look at the previous two questions.)

A. 0001102

B. 0101102

**C. 0101002**

D. 0100102

E. 0101112

F. 1101102

1. Change the representation of 10102 from base two to base ten.

A. 810

B. 910

**C. 1010**

D. 1110

E. 1210

F. 1410

2. What number goes in the empty cell of the table:

Power of 2 6 5 4 3 2 1 0

Decimal ???? 32 16 8 4 2 1

A. 34

B. 48

C. 60

**D. 64**

E. 68

F. 72

3. Change the representation of 1000102 from base two to base ten.

A. 14

B. 28

C. 32

**D. 34**

E. 46

F. 52

4. Change the representation of 0000 11112 from base two to base ten. (Hint: you should get this instantly.)

A. 6

B. 7

C. 12

**D. 15**

E. 16

F. 31

5. Change the representation of 0011 11002 from base two to base ten. (Hint: use your previous answer.)

A. 14

B. 28

C. 32

**D. 60**

E. 64

F. 72

6. Say that you are using unsigned binary to represent integers with 6 bits. What range of integers can be represented?

A. 0 to 64

B. 1 to 64

**C. 0 to 63**

D. 1 to 128

E. 0 to 255

F. 1 to 247

7. Change the representation of A516 from base sixteen to base ten.

A. 46

**B. 165**

C. 232

D. 245

E. 305

F. 1025

8. Here is a number represented in base 16 notation: 5A3F. Write the number in unsigned binary notation.

**A. 0101 1010 0011 1111**

B. 1111 0011 1010 1001

C. 1100 1000 0111 1111

D. 1010 1001 0011 1000

E. 1011 0001 0000 1110

F. 1110 1011 1011 1001

9. Convert the representation of the following from base 16 to base 8: 0x37A.

A. 011 111 1010

B. 0772

C. 001 101 111 010

**D. 1572**

E. 8214

F. 7426

10. Represent 2710 in base 2.

Algorithm: Convert a number

from base 10 to base B repn

place = 0;

number = number to be converted

while (number > 0 )

{

digit[place] = number mod B ;

number = number div B ;

place = place + 1 ;

}

A. 101101

B. 10110

C. 1011

**D. 11011**

E. 101101

F. 100110

11. Convert 3045 from base 5 to base 10.

A. 15

B. 23

C. 27

D. 32

**E. 79**

F. 83

12. Convert 3045 from base 5 to base 2 (use your previous answer).

A. 1100 1001

B. 0110 0100

C. 0010 1000

D. 1000 1100

E. 1010 1011

**F. 0100 1111**

13. Write the unsigned binary number 0010 1110 in hexadecimal representation

A. 0232

B. 2F

C. 3E

D. 07

**E. 2E**

F. FF

14. Write the unsigned binary number 000 101 110 in octal representation.

A. 022

B. 023

C. 013

D. 122

**E. 056**

F. 023

15. Write the unsigned binary number 000 101 110 in decimal representation.

A. 26

**B. 46**

C. 52

D. 64

E. 122

F. 131

1. In the following one-bit wide addition, what are the result R and the carry, C?

C0

0

1

--

R

A. C=0; R=0

**B. C=0; R=1**

C. C=1; R=0

D. C=1; R=1

2. In the following one-bit wide addition, what are the result R and the carry, C?

C1

0

1

--

R

A. C=0; R=0

B. C=0; R=1

**C. C=1; R=0**

D. C=1; R=1

3. Which one of the following is done correctly?

**A.**

**0100**

**0110**

**0101**

**----**

**1011**

B.

0100

0110

0101

----

1111

C.

0110

0110

0101

----

1111

D.

0101

0110

0101

----

1001

4. Which one of the following is done correctly?

A.

0100

1110

1001

----

1001

B.

0000

1110

1001

----

0111

C.

0111

1110

1001

----

1001

**D.**

**1000**

**1110**

**1001**

**----**

**0111**

5. Perform the following addition:

1100

0101

----

**A. 0001 with a carry out of the left column of 1**

B. 0001 with a carry out of the left column of 0

C. 1001 with a carry out of the left column of 1

D. 1110 with a carry out of the left column 1

6. Here is a two's complement representation of an integer:

0011 1001

What is the two's complement representation of the negation of the integer?

A. 1100 0110

B. 1011 1001

C. 1100 1110

**D. 1100 0111**

7. Here is a two's complement representation of an integer:

1100 0111

What is the two's complement representation of the negation of the integer?

A. 1100 0110

**B. 0011 1001**

C. 0000 1110

D. 0100 0111

8. Here is a correctly performed addition:

1100

0100

1110

----

0010

What is true about overflow for this addition?

A. If the operands are regarded as unsigned binary, then the result shows no overflow.

If the operands are regarded as two's complement binary, then the result shows no overflow.

**B. If the operands are regarded as unsigned binary, then the result shows overflow.**

**If the operands are regarded as two's complement binary, then the result shows no overflow.**

C. If the operands are regarded as unsigned binary, then the result shows no overflow.

If the operands are regarded as two's complement binary, then the result shows overflow.

D. If the operands are regarded as unsigned binary, then the result shows overflow.

If the operands are regarded as two's complement binary, then the result shows overflow.

9. Here is a correctly performed addition:

1000

1100

1010

----

0110

What is true about overflow for this addition?

A. If the operands are regarded as unsigned binary, then the result shows no overflow.

If the operands are regarded as two's complement binary, then the result shows no overflow.

B. If the operands are regarded as unsigned binary, then the result shows overflow.

If the operands are regarded as two's complement binary, then the result shows no overflow.

C. If the operands are regarded as unsigned binary, then the result shows no overflow.

If the operands are regarded as two's complement binary, then the result shows overflow.

**D. If the operands are regarded as unsigned binary, then the result shows overflow.**

**If the operands are regarded as two's complement binary, then the result shows overflow.**

10. Say that there are two operands represented using the two's complement method:

operand A = 0011 1010 operand B = 0110 1011

Which of the following uses of the binary addition algorithm shows the problem A - B?

A.

0011 1010

0110 1011

---------

B.

1100 0110

0110 1011

---------

**C.**

**0011 1010**

**1001 0101**

---------

D.

1100 0101

1001 0110

---------

1. What is a source file?

**A. a text file that contains statements of programming language**.

B. a subdirectory that contains programs.

C. a file that contains data for a program.

D. a document that contains the requirements for a project.

2. What is a register?

A. a part of the computer system that keeps track of system parameters.

**B. a part of the processor that holds a bit pattern.**

C. a part of the processor that contains its unique serial number.

D. the part of the system bus that contains data.

3. What character, in SPIM assembly language, starts a comment?

**A. #**

B. $

C. //

D. \*

4. How many bits are there in each MIPS machine instruction?

A. 8

B. 16

**C. 32**

D. Different instructions are diferent lengths.

5. When you open a source file from the File menu of SPIM, what two things happen?

A. The file is loaded into memory and execution starts.

B. SPIM is booted and the file is opened in the editor.

**C. The file is assembled into machine instructions, and the machine instructions are loaded into SPIM's memory.**

D. The program is run and the results are saved to disk.

6. What is the program counter?

A. a register that keeps counts the number of errors during execution of a program.

B. a part of the processor that contains the address of the first word of data.

C. a variable in the assembler that numbers the lines of the source file.

**D. a part of the processor that contains the address of the next machine instruction to be fetched.**

7. Say that you push F10 to execute one instruction. What amount is added to the program counter?

A. 1

B. 2

**C. 4**

D. 8

8. What is a directive, such as the directive .text?

A. an assembly language statement that results in one machine language instruction.

B. one of the menu choices in the SPIM menu system.

C. a machine language instruction that causes an operation on data.

**D. a statement that tells the assembler something about what the programmer wants, but does not itself directly correspond to a machine instruction.**

9. What is a symbolic address?

A. a location in memory containing symbolic data.

B. a byte in memory that holds the address of data.

C. the symbol given as the argument for a directive.

**D. a name used in assembly language source code for a location in memory.**

10. At what address does the SPIM simulator put the first machine instruction when it is running with the Bare Machine option turned ON?

A. 0x00000000

**B. 0x00400000**

C. 0x10000000

D. 0xFFFFFFFF

1. When a register is loaded what happens?

A. The bits of the register are set to all ones.

B. The bit pattern in the register is copied to a location in memory.

C. A bit pattern at a memory location is copied to the register. The memory is set to all zeros.

**D. A bit pattern at a memory location is copied to the register. Memory is not changed.**

2. When a register is stored what happens?

A. The bits of the register are set to all ones.

**B. The bit pattern in the register is copied to a location in memory.**

C. A bit pattern at a memory location is copied to the register. The memory is set to all zeros.

D. A bit pattern at a memory location is copied to the register. Memory is not changed.

3. By software convention, the machine instructions of a program are put in a designated section of memory. What is this section called?

A. Data segment.

B. Stack segment.

C. Program segment.

**D. Text segment.**

4. What general purpose register is permanently set to thirty-two zero bits?

**A. $0**

B. $1

C. $31

D. $32

5. What part of the processor chip performs arithmetic and logical operations?

A. CPU

**B. ALU**

C. ROM

D. PCI

6. Where do the operands for an arithmetic machine instruction come from?

A. Both operands are registers.

B. Both operands come from memory.

C. One operands must be a register, the other one may be memory or a register.

**D. One operand must be a register, the other may be a register or may be part of the machine instruction.**

7. What is an opcode?

A. The part of a machine instruction that designates the data to be used.

**B. A bit field, part of a machine instruction, that designates a machine operation.**

C. The part of the processor chip that performs decoding operations.

D. The part of a machine instruction that is used as data in an operation.

8. There are 32 general purpose registers. So a machine instruction must use a bit field of what size to designate a register?

A. 3

B. 4

**C. 5**

D. 16

9. What is the mnemonic name of a register?

A. A register number like $0 or $31.

B. A name that helps you remember the hardware characteristics of the register.

C. The bit pattern that designates the register in a machine instruction.

**D. A name like $s0 that helps you remember the conventional software uses for the register.**

10. What, in order, are the three steps of the machine cycle?

**A. fetch, increment, execute.**

B. Execute, fetch, increment.

C. Fetch, exectue, increment.

D. increment, fetch, execute.

1. Some machine instructions contain one of the operands for the operation they specify. What is such an operand called?

**A. immediate operand**

B. embedded operand

C. binary operand

D. machine operand

2. What is it called when a logical operation is performed between the bits of each column of the operands to produce a result bit for each column?

A. logic operation

**B. bitwise operation**

C. binary operation

D. column

3. When an operation is actually performed, what is true of the operands in the ALU?

A. At least one operand must be 32 bits wide.

B. Each operand can be any size.

C. Both operands must come from registers.

**D. They must each be 32 bits wide.**

4. Sixteen bits of data from an ori instruction are used as an immediate operand. During execution, what must first be done?

A. The data is zero extended by 16 bits on the right.

**B. The data is zero extended by 16 bits on the left.**

C. Nothing needs to be done.

D. Only 16 bits are used from the other operand.

5. What is it called with a bit pattern is copied into a register?

**A. The register is loaded with the pattern.**

B. The register is stuffed with the pattern.

C. The register is filled with the pattern.

D. The register is set with the pattern.

6. Which of the following instructions loads register $5 with the bit pattern that represents positive 4810?

A. ori $5,$0,0x48

B. ori $5,$5,0x48

**C. ori $5,$0,48**

D. ori $0,$5,0x48

7. Can the ori instruction put the two's complement representation of a negative integer into a register?

**A. No.**

B. Yes.

8. Which of the following instructions clears all the bits in register $8 except for the low order byte, which is unchanged?

A. ori $8,$8,0xFF

B. ori $8,$0,0x00FF

C. xori $8,$8,0xFF

**D. andi $8,$8,0xFF**

9. What is the result of performing an exclusive or of a bit pattern with itself?

**A. The result is all zero bits.**

B. The result is the same as the original.

C. The result is all one bits.

D. The result is the reverse of the original.

10. Do all machine instructions have the same parts?

**A. No. Different types of machine instructions are made of different fields.**

B. No. Each machine instruction is completely different from any other.

C. Yes. All machine instructions have the same parts in the same order.

D. Yes. All machine instructions have the same parts, but they might be in different orders.

1. Shift left logical the following bit pattern by one position:

0011 1111

**A. 0111 1110**

B. 0111 1111

C. 0011 1110

D. 1011 1111

2. Shift left logical the following bit pattern by two positions:

0011 1111

A. 0111 1111

B. 1111 1110

**C. 1111 1100**

D. 1011 1111

3. Shift right logical the following bit pattern by one position:

0011 1111

A. 0000 1111

B. 1111 1110

C. 0001 1110

**D. 0001 1111**

4. Here is a program that loads register $5 with a bit pattern. Complete the program so that register $4 receives the pattern in register $5 shifted left logical by three positions.

ori $5, $0, 0x92AF # put a bit pattern into register $5

sll \_\_\_, \_\_\_, \_\_\_ # shift left logical by three, put

# result in register $4

A. sll $5, $3, $4

B. sll 3, $5, $4

C. sll 5, 3, 4

**D. sll $4, $5, 3**

5. What is the bit-wise OR of the following patterns:

0110 1101

0100 1010

**A. 0110 1111**

B. 0111 0110

C. 0110 1001

D. 1110 0110

6. What is the bit-wise AND of the following patterns:

0110 1101

0100 1010

A. 0110 0110

**B. 0100 1000**

C. 0110 1001

D. 1110 0110

7. What is the bit-wise XOR of the following patterns:

0110 1101

0100 1010

A. 0110 0110

**B. 0010 0111**

C. 0110 1001

D. 1110 0110

8. What is the bit-wise NOR of the following patterns:

0110 1101

0100 1010

A. 0110 1111

**B. 1001 0000**

C. 1110 1011

D. 1110 1110

9. Write the assembly language statement that will reverse the values of each bit in register $5 and put the result in register $8.

A. nori $8,$5,$0

**B. nor $8,$5,$0**

C. xor $8,$5,$0

D. nor $5,$8,$0

10. Write the assembly language instruction that copies the bit pattern in register $13 to register $15.

A. ori $15,$13,$0

B. andi $13,$15,$15

C. and $13,$15,$15

**D. or $15,$13,$0**

1. How many bits are the operands of the ALU?

**A. Always 32.**

B. 8, 16, or 32

C. 32 or 64

D. Any number of bits up to 32.

2. What procedure does the addu instruction call for?

**A. The binary addition algorithm.**

B. The unsigned addition algorithm.

C. The bit-wise addition algorithm.

D. The universal addition algorithm.

3. What type of data may be in the registers used as operands for the addu instruction?

A. 32-bit unsigned in both registers.

B. 32-bit two's complement in both registers.

**C. 32-bit unsigned or 32-bit two's complement, the same type in each register.**

D. 32-bit unsigned or 32-bit two's complement, either type in either register.

4. What is a trap?

**A. A trap is an interruption in the normal machine cycle.**

B. A trap is an instruction that captures data.

C. A trap is a shortened version of a normal 32-bit instruction.

D. A trap is a bus signal that says that something is wrong.

5. Write the instruction that adds the contents of registers $8 and $9 and puts the result in register $13.

A. add $8,$9,$13

B. addi $8,$9,$13

**C. addu $13,$8,$9**

D. add $9,$8,$13

6. Can the immediate operand of an addiu instruction be a two's complement negative integer?

A. No. The immediate operand is zero-extended to 32 bits.

**B. Yes. When the instruction is executed, the immediate operand is sign-extended to 32 bits.**

C. No. A 16-bit immediate operand is too small for two's complement.

D. Yes. Immediate operands are always two's complement.

7. What is it called when bit 15 of a 16-bit immediate operand is copied to the 16 bits to its left to form a 32-bit operand?

**A. Sign extention**

B. Bit extention

C. Zero extention

D. Immediate extention

8. Write an instruction that adds the value 12 to the value in register $6. Use an instruction which ignores overflow.

A. add $6,12

B. addi $6,$0,12

**C. addiu $6,$6,12**

D. addi $6,12

9. What integer subtraction instruction uses two operand registers, one result register, and does not trap if overflow is detected?

**A. subu**

B. sub

C. subi

D. subiu

10. Which instruction loads register $17 with a -99?

**A. addiu $17,$0,-99**

B. sub $17,$0,-99

C. addiu $17,-99

D. subu $17,$17,99

1. If you have two integers, each represented in 32 bits, how many bits might be needed to hold the product?

A. 16

B. 32

**C. 64**

D. 128

2. What are the names of the two registers that hold the result of a multiply operation?

A. high and low

**B. hi and lo**

C. R0 and R1

D. $0 and $1

3. Which operation is used to multiply two's complement integers?

**A. mult**

B. multu

C. multi

D. mutt

4. Which instruction moves the least significant bits of a product into register eight?

A. move $8,lo

B. mvlo $8,lo

**C. mflo $8**

D. addu $8,$0,lo

5. If you have two integers, each represented in 32 bits, how many bits should you be prepared to have in the quotient?

A. 16

**B. 32**

C. 64

D. 128

6. After a div instruction, which register holds the quotient?

**A. lo**

B. hi

C. high

D. $2

7. What instruction is used to divide two's complement integers?

A. dv

B. divide

C. divu

**D. div**

8. Perform an arithmetic shift right by two bits of the following bit pattern:

1001 1011

**A. 1110 0110**

B. 0010 0110

C. 1100 1101

D. 0011 0111

9. If a general purpose register holds a bit pattern that represents an integer, an arithmetic shift right of one position has what effect?

A. If the integer is unsigned, the shift divides it by two. If the integer is signed, the shift divides it by two.

B. If the integer is unsigned, the shift divides it by two. If the integer is signed, the shift may produce an incorrect result.

**C. If the integer is unsigned, the shift may produce an incorrect result. If the integer is signed, the shift divides it by two.**

D. The shift multiplies the number by two.

10. Which list of instructions computes 3x+7, where x starts out in register $8 and the result is put in $9?

**A.**

**ori $3,$0,3**

**mult $8,$3**

**mflo $9**

**addiu $9,$9,7**

B.

ori $3,$0,3

mult $8,$3

addiu $9,$8,7

C.

ori $3,$0,3

mult $8,$3

mfhi $9

addiu $9,$9,7

D.

mult $8,3

mflo $9

addiu $9,$9,7

1. What operation copies data from main memory into a general purpose register?

**A. load**

B. store

C. move

D. add

2. Which one of the following addresses is word aligned?

A. 0x01234567

**B. 0x00FA0700**

C. 0x77000003

D. 0x00000042

3. Say that four bytes in main storage contain bits that represent a 32-bit integer. What is the address is used for this integer?

A. The address of the byte with the highest address of the four.

B. The address of each byte is used.

C. Only one byte at a time can be addressed.

**D. The address of the byte with the lowest address of the four.**

4. Here is a 32-bit pattern: 0x00224477. This pattern is to be stored in main memory using bytes at addresses 0x10000000, 0x10000001, 0x10000002, and 0x10000003. On a big endian processor, what bit pattern is contained in address 0x10000000?

**A. 0x00**

B. 0x22

C. 0x44

D. 0x77

5. A lw is to load register $5 from location 0x0040000C in memory. Register $10 contains 0x00400000. Write the assembly language instruction:

A. lw $10,0x0C($10)

B. lw $10,0x0C($5)

**C. lw $5,0x0C($10)**

D. lw $5,0x0C(400000)

6. Register $10 contains 0x10000000. Beginning at that address there are five integers in a row. Write the instruction that loads the last integer into register $7.

A. lw $7, 50($10)

B. lw $7, 20($10)

**C. lw $7, 16($10)**

D. lw $7, 40($10)

7. Register $5 contains the address 0x10000100. Write the instruction that loads the four bytes that precede this address into register $7.

A. lw $7, 4($5)

**B. lw $7, -4($5)**

C. lw $5, 4(-$5)

D. lw $7, 0($5-4)

8. Write the assembly instruction that fills register $10 with 0x10000000

**A. lui $10, 0x1000**

B. lui $10, 0x10000000

C. ori $10, $0, 0x10000000

D. ori $10, $10, 0x1000

9. Say that somehow register $10 has been loaded with the address 0x10000000. Write the instruction that alters $10 so that it contains 0x100000F0.

A. ori $10, $0, 0x00F0

B. or $10, $10, 0x00F0

**C. ori $10, $10, 0x00F0**

D. andi $10, $10, 0x00F0

10. Examine the following program fragment.

## fragment.asm

.text

.... program statements ...

.data

value: .word 23

result: .word 97

Assuming that SPIM starts the data section at address 0x10000000, what address does symbolic address result represent?

**A. 0x10000004**

B. 0x10000000

C. 0x10000003

D. 0x00000097

11. Refer back to the previous fragment. We want register $8 to contain the address of the first byte in the data section. What instruction does this?

A. lui $8,1000

B. lui $8,$8,0x1000

C. lui $8,0x10000000

**D. lui $8,0x1000**

12. Refer back to the previous fragment. Assume that register $8 contains the address 0x10000000. What instruction stores the contents of register $4 into location result?

**A. sw $4,4($8)**

B. sw $4,0($8)

C. sw $4,0x0($8)

D. sw $4,$0x05($8)

1. What is the smallest addressable unit of main memory?

**A. byte**

B. bit

C. nibble

D. halfword

2. Which of the following instructions does sign extension?

A. lbu

**B. lb**

C. add

D. lhu

3. Say that:

Memory at 0x10000000 contains 0x80

Register $5 contains 0x10000000

What is put in register $8 after lb $8,0($5) is executed?

A. 0x88888880

B. 0x00000080

C. 0x80000000

**D. 0xFFFFFF80**

4. What instruction is used to store a byte to memory?

**A. sb**

B. sbu

C. lb

D. sw

5. Say that:

Memory at 0x10000000 contains 0x80

Memory at 0x10000001 contains 0x00

Register $5 contains 0x10000000

Say that the MIPS chip is running in little-endian mode (as does SPIM on an Intel computer). What is put in register $8 after lh $8,0($5) is executed?

A. 0xFFFFFF80

B. 0x88888880

**C. 0x00000080**

D. 0x80000000

6. Say that:

Memory at 0x10000000 contains 0x80

Memory at 0x10000001 contains 0x00

Register $5 contains 0x10000000

Say that the MIPS chip is running in big-endian mode (as does SPIM on an Apple computer). What is put in register $8 after lh $8,0($5) is executed?

**A. 0xFFFF8000**

B. 0xFFFFFF80

C. 0x00000080

D. 0x80000000

7. Which one of the following address are half-word aligned?

A. 0x01004F35

B. 0x01004F37

**C. 0x01004F3A**

D. 0x01004F3F

8. Say that data is in memory and the base register has been initialized correctly. You have the following program:

lh $5,0($10)

lb $6,4($10)

addu $7,$5,$4

What does the addu instruction do?

**A. It performs the binary addition algorithm on whatever 32-bit patterns are in registers $4 and $5.**

B. It performs a 16-bit addition because that is the size of the largest operand.

C. It performs an 8-bit addition.

D. The instruction causes a trap because the operands are not the same sizes.

9. Which of the following assembler directives reserves 1210 bytes of memory?

A. .word 3

B. .byte 12

C. .block 6

**D. .space 12**

10. You wish to speed up the execution of a C program. The program runs on a 32-bit processor. You notice that the variables in the program are a mix of short int, int and long int variables. The program does a great deal of integer arithmetic. How might you speed up this program?

**A. Make as many variables of type int as is possible.**

B. Make as many variables of type short int as is possible.

C. Make all variables as small as is needed for the range of values they are exected to hold.

D. Shorten the names of all the variables.

11. A digital image is stored in a file. The pixels of the image represent a gray level of 0 to 255. What instruction are you likely to use in loading a register with the value of a pixel?

A. lb

**B. lbu**

C. lh

D. lhu

12. How does SPIM display the data section of simulated main memory?

A. One byte per address in columns.

B. In groups of 4-byte words with the highest address on the right.

**C. In groups of 4-byte words with the lowest address on the right.**

D. This depends on the type of data in memory.

1. What are the three steps in the machine cycle?

A. increment the PC; fetch the instruction; execute the instruction

B. fetch the instruction; execute the instruction; increment the PC

C. execute the instruction; fetch the instruction; increment the PC

**D. fetch the instruction; increment the PC; execute the instruction**

2. What are the four bytes immediately following a jump instruction called?

A. fetch delay slot

B. pipeline delay slot

**C. branch delay slot**

D. PC advance slot

3. What is a pipeline?

A. Several words of data from memory are moved into the processor before instructions need them.

**B. Several sequential instructions are simultaneously prepared for execution while one instruction finishes its execution.**

C. A single instruction is divided into four phases and each phase is executed in one machine cycle.

D. Multiple items of data are sent down the system bus like water in a pipe.

4. Say that a sll instruction is located in memory at address 0x400100, and an add instruction is located in memory at address 0x400104. After the add instruction executes, what value will be in the PC?

A. 0x400100

B. 0x400104

C. 0x400105

**D. 0x400108**

5. Say that a j (jump) instruction is located in memory at address 0x400100, and a sll instruction is located in memory at address 0x400104. After the j instruction executes, what value will be in the PC?

A. 0x400100

B. 0x400101

C. 0x400102

**D. 0x400104**

6. Here is a schematic program loop.

Address Instruction

(details omitted) PC just after this

instruction has executed

(at the bottom of the cycle)

............... ........... 00450008

00450008 add 0045000C

0045000C store 00450010

00450010 jump 0x00450008 004500\_\_\_

00450014 no-op 004500\_\_\_

What numbers go into the two blanks?

**A.**

**14**

**08**

B.

14

00

C.

00

08

D.

14

18

7. Here is a 32-bit j instruction. The first 6 bits are the op-code.

000010 00 0001 0000 0000 0000 0000 1000

Here is the value of the PC while the target address is being constructed:

0000 1000 0001 0000 0000 1100 0110 1000

What address does the j put into the PC?

**A. 0000 00 0001 0000 0000 0000 0000 1000 00**

B. 0000 1000 0001 0000 0000 1100 0110 1000

C. 0000 10 0001 0000 0000 1100 0110 1000 00

D. 1000 00 0001 0000 0000 0000 0000 1000 00

8. Examine the following program fragment. The program is to add $5 and $6 together only if they are not equal.

ori $5,$0,8 # load $5 with 8

ori $6,$0,9 # load $6 with 9

\_\_\_\_ $5,$6,spot

\_\_\_\_ $0,$0,0 # branch delay slot

addu $8,$5,$6 # $8 = $5 + $6

spot:

Pick instructions to fill the blanks.

A. beq ; addu

B. bne ; sll

C. bne ; addu

**D. beq ; sll**

9. Here is an if-then-else structure. The code is to compare $10 and $11. If these registers contain the same bit pattern, set register $7 to 1. Otherwise set $7 to 0.

ori $10,$0,123

ori $11,$0,123

\_\_\_ $10,$11,\_\_\_\_\_

sll $0,$0,0

ori $7,$0,0

j \_\_\_\_\_

sll $0,$0,0

equal: ori $7,$0,1

join: .....

Which choices should fill the blanks?

A. bne ; equal ; join

B. beq ; join ; equal

**C. beq ; equal ; join**

D. bne ; join ; equal

10. Say that registers $5 and $6 each contain an ASCII character in the low order byte. Can the beq instruction be used to compare the characters?

A. Yes, because beq will recognize the character data and do a character comparison.

B. No, because beq only works with two's complement integers.

C. No, because beq only works with full 32-bit data.

**D. Yes, because beq compares bit patterns regardless of what they represent.**

1. Examine the following program fragment:

ori $8,$0,13

ori $9,$0,1

bltz $8,target

sll $0,$0,0

ori $9,$0,0

target: sll $0,$0,0 # arbitrary instruction

What value is found in $9 when control reaches target?

**A. 0**

B. 1

C. 4

D. 13

2. Trick Question: Examine the following program fragment:

ori $8,$0,-57

ori $9,$0,1

bltz $8,target

ori $9,$0,0 # think about the delay

# slot

target: sll $0,$0,0 # arbitrary instruction

What value is found in $9 when control reaches target?

**A. 0**

B. 1

C. 3

D. 4

3. Examine the following program fragment:

ori $8,$0,13

ori $9,$0,1

bgez $8,target

sll $0,$0,0

ori $9,$0,0

target: sll $0,$0,0 # arbitrary instruction

What value is found in $9 when control reaches target?

A. 0

**B. 1**

C. 4

D. 13

4. Examine the following program fragment (slightly different from the previous):

ori $8,$0,13

bgez $8,target

ori $9,$0,1

ori $9,$0,0

target: sll $0,$0,0 # arbitrary instruction

What value is found in $9 when control reaches target?

A. 0

**B. 1**

C. 4

D. 13

5. Examine the following program fragment:

addiu $3,$0,-13

addiu $7,$0,23

?????

Pick the instruction to replace ????? that will set register $10 to one.

A. sltu $3,$7,$10

B. slt $10,$7,$3

**C. slt $10,$3,$7**

D. sltu $10,$3,$7

6. Examine the following program fragment:

addiu $3,$0,-13

slti $5,$3,-8

What value is in $5 after both instructions exectute?

A. 0

**B. 1**

C. -8

D. -13

7. Examine the following program fragment:

ori $3,$0,25

slti $5,$3,53

What value is in $5 after both instructions exectute?

A. 0

**B. 1**

C. 25

D. 53

8. (Very Tricky:) Examine the following program fragment:

addiu $3,$0,-1

slti $5,$3,17

What value is in $5 after both instructions exectute? (If your answer is incorrect, run the program with SPIM and examine the registers. SPIM does not "know" how the bit pattern got into $3, it's just a pattern and the slti instruction acts on it mechanically.)

**A. 0**

B. 1

C. -8

D. -13

9. Which style of implementing a counting loop is usually easiest to understand?

A. data driven loop

B. bottom driven loop

C. conditional driven

**D. top driven loop**

10. Examine the following program fragment:

ori $5,$0,5 # initialize count

ori $8,$0,0 # initialize accumulator

test: bltz $5,done

sll $0,$0,0

addu $8,$8,$5 # add count to accumulator

addiu $5,$5,-1

j test

sll $0,$0,0

done: sll $0,$0,0

How many times is the addu instruction executed?

A. 0

B. 5

**C. 6**

D. 7

1. Some microprocessors have a 64-bit word size. Can these microprocessors compute things that 32-bit microprocessors can not?

**A. No. All microprocessors have the fundamental operations it takes to have equal computing power.**

B. No. Two 32-bit microprocessors can be wired together to get the same power as a 64-bit microprocessors.

C. Yes. They can compute with much larger numbers than microprocessors with smaller word sizes.

D. Yes. They can have many more machine operations by using 64-bit machine instructions.

2. Is speed considered to be a part of computing power?

A. No — since the speed of a microprocessor is easily affected by the rest of the computer system.

**B. No — Computing power is concerned only with what can be computed, not how long it takes.**

C. Yes — faster microprocessors have more computing power.

D. Yes — but if two microprocessors have equal speed, the one with the most instructions is more powerful.

3. Does a microprocessor need special instructions for input/output and for graphics?

**A. No. This is done by using ordinary load and store operations with special addresses that have been assigned to the devices.**

B. No. This is done by loading a storing special registers within the microprocessor.

C. Yes. Special I/O and graphics machine instructions are used, but these are not included in the definition of computing power.

D. Yes. Special I/O and graphics machine instructions are used, and these are included in the definition of computing power.

4. How much computing power can be expected of a modern microprocessor?

A. The more money a microprocessor costs, the more computing power it will have.

B. Microprocessors show great increases in computing power every year, so more recent ones have more power than old ones.

C. Within the same family, microprocessors have the same computing power. But microprocessors in different families cannot be compared.

**D. All past and present general purpose microprocessors are equal in computing power.**

5. What is throughput?

A. ... another word for computing power.

B. ... how fast a computer system runs.

**C. ... how much computing a computer system can perform in a unit of time.**

D. ... how much data a mass storage system can store.

6. What does RISC stand for?

A. Regularized Instruction System Chip

B. Reduced Information System Computing

C. Registers Implemented with Silicon Chips

**D. Reduced Instruction Set Computer**

7. In structured programming, what is a block?

**A. A block is a section of code with just one entry point and just one exit point.**

B. A block is a section of code with one or more entry points and just one exit point.

C. A block is a section of code with one or more entry points and one or more exit points.

D. A block is a sequential section of code.

8. Which one of the following statements is true?

A. Code blocks in sequence are not structured.

B. Programing done in assembly language is automatically structured.

**C. Two or more code blocks in sequence are structured.**

D. Object oriented languages are not structured.

9. Is the alternation of code blocks structured?

A. No.

**B. Yes.**

10. Is it possible to write a program in assembly language that can compute something that can't be computed by a program in a structured language?

**A. No. Unstructured assembly language and structured high level language have equal computational power.**

B. No. But the high level language can compute many more things than can be done in assembly language.

C. Yes. Assembly language gives access to many operations not possible in any high level language.

D. Yes. Assembly language has many more control structures and therefore more computational power.

1. How many bytes are used for an ASCII-encoded character?

**A. 1**

B. 2

C. 4

D. 8

2. Which of the following puts a null-terminated string "Hello World" in memory?

A. .ascii "Hello World"

B. .text "Hello World"

C. .word "Hello World"

**D. .asciiz "Hello World"**

3. How is a null-terminated string arranged in memory?

A. Characters are grouped four to a full word from right to left. Extra space at the end is filled with zeros.

**B. Characters are put in sequential order with a null byte after the last character.**

C. Characters are put in sequential order one per word with a word of null after the last character.

D. Characters are put in sequential order in memory. Space characters are replaced with null bytes.

4. A character has been loaded into a register by using a lbu instruction. What does the register look like?

**A. The character is in the low-order byte. The three high-order bytes are zero.**

B. The character is in the low-order byte. The three high-order bytes are whatever they were just before the instruction.

C. The character is in the high-order byte. The three low-order bytes are zero.

D. The character is in the low-order byte. The three high-order bytes contain one-bits.

5. How is a character pointer typically moved from one character to the next?

A. It is incremented by four with a addiu instruction.

B. It is incremented by one with a add instruction.

**C. It is incremented by one with a addiu instruction.**

D. It is changed with a move instruction.

6. How is an array of integers typically implemented?

A. The integers of the array are put in sequential words of memory. A word of zeros follows the last integer.

**B. The integers of the array are put in sequential words of memory. Another word of memory contains the length of the array.**

C. The integers of the array are put in sequential bytes of memory. The last byte holds the length of the array.

D. Each integer of the array is assigned to one of the general purpose registers.

7. What instruction can be used to load register $10 with the first address of the .data section?

**A. lui $10,0x1000**

B. lui $10,0x10000000

C. ori $10,0x1000

D. andi $10,0x4000

8. How much should a base register be incremented by to move from one integer to the next in an array of integers?

A. 1

B. 2

**C. 4**

D. 8

9. Say that register $8 contains an integer and that register $9 contains a sum. The integer is to be added to the sum only if it is positive. Which of the following code sequences does this?

**A.**

**bltz $8,noadd**

**sll $0,$0,0**

**addu $9,$9,$8**

**noadd:**

B.

bltz $8,noadd

addu $9,$9,$8

noadd:

C.

blgez $8,noadd

sll $0,$0,0

addu $9,$9,$8

noadd:

D.

slt $5,$8,$0

beq $5,$0,noadd

sll $0,$0,0

addu $9,$9,$8

noadd:

10. When the SPIM simulator is set to "bare machine" to what value should the PC be initilized?

A. 0x00000000

B. 0x10000000

C. 0x100000

**D. 0x400000**

1. Register $8 is the first register conventionally used to hold temporary values. What is its mnemonic name?

A. $temp

**B. $t0**

C. $t1

D. $t8

2. What is the mnemonic name of register $0?

A. zero

**B. $zero**

C. $z0

D. $t0

3. Does the extended assembler translate assembly language into machine instructions not available to the basic assembler?

**A. No. Extended assembly language is just an alternate syntax used to specify the same machine instructions.**

B. No. The extended assembler uses many fewer of the actual machine instructions.

C. Yes. The extended assembler outputs instructions for the extended part of the processor.

D. Yes. The extended assembler is the only way to ask for floating-point operations.

4. What is true just after the return from a call to a subroutine?

A. The temporary registers are the same as they were before the call, and the saved registers may have been altered.

B. The saved registers are the same as they were before the call, and the temporary registers have certainly been altered.

C. Values in the saved registers are safely stored in memory, but values in the temporary registers have not been saved.

**D. The saved registers are the same as they were before the call, and the temporary registers may have been altered.**

5. What is the most fundamental part of an operating system?

A. The shell.

B. The peel.

**C. The kernel**.

D. The seed.

6. Translate the following pseudoinstruction into basic assembly: move $t8,$t3

A. addu $t3,$t8,$t3

B. ori $t8,$0,$t3

C. ori $t8,$t3

**D. addu $t8,$0,$t3**

7. Here is the load immediate pseudoinstruction: li d,value. What is true of value?

A. It may be any integer, positive or negative, that fits into 16 bits.

**B. It may be any integer, positive or negative, that fits into 32 bits.**

C. It may be any positive integer that fits into 32 bits.

D. It may be any positive integer that fits into 16 bits.

8. How is the li d,value translated by the extended assembler?

A. It is always translated into the addiu basic instruction.

B. It is always translated into the one basic instruction.

**C. How it is translated depends on the size and sign of the value.**

D. It is always translated into two basic instructions.

9. Inspect the following code:

.data

numa: .word 23

numb: .word 99

numc: .word -12

Which instruction puts the address of the word that contains 99 into register $s3?

**A. la $s3,numb**

B. lw $s3,numb

C. lw $s3,99

D. la $s3,4(numa)

10. Inspect the following code (the same as above):

.data

numa: .word 23

numb: .word 99

numc: .word -12

Which instruction stores the contents of register $s3 into memory at numc ?

A. sa $s3,numc

**B. sw $s3,numc**

C. sw numc,$s3

D. move $s3,numc

1. What is the name for the small collection of services on SPIM that are invoked using a syscall instruction?

A. wolf trap

**B. exception handler**

C. BIOS

D. operating system

2. Does the syscall instruction correspond to just one machine language instruction?

A. No—it is so complicated that it takes several machine instructions.

B. No—it is special and does not correspond to any machine instructions.

**C. Yes—it is a fundamental operation of the hardware.**

D. Yes—all assembly language statements correspond to one machine instruction.

3. Which of the following fragments correctly writes "Hello" to the monitor?

A.

lw $v0,4

lw $a0,hello

syscall

. . .

hello: .asciiz "Hello"

**B.**

**li $v0,4**

**la $a0,hello**

**syscall**

**. . .**

**hello: .asciiz "Hello"**

C.

li $v0,4

lw $a0,hello

syscall

. . .

hello: .ascii "Hello"

D.

li $v0,4

li $a0,"Hello"

syscall

. . .

4. Where is the integer after the "read integer" service returns control?

A. In the input buffer.

B. In a register specified by the user.

**C. In register $v0.**

D. In register $a0.

5. What parameters are expected for the "read string" service?

A. The address of the input buffer.

B. The length of the input buffer.

C. The addresses of the input buffer and of the output buffer.

**D. The length of the input buffer and the address of the input buffer.**

6. Are the code numbers for the various services determined by the MIPS hardware or by the software of the the exception handler?

A. This is a hardware feature of the MIPS chip.

B. This is a hardware feature of the various IO devices attached to the computer.

C. Software—the user program sets up the service codes before it starts execution.

**D. Software—the exception handler software examines the code and branches to the indicated service.**

7. In an actual computer system with a full-scale operating system, what would the exit service do?

**A. Return control to the operating system.**

B. Halt the processor.

C. Halt the operating system.

D. Immediately start running another user program.

8. Is "backspace" a character?

A. No — hitting backspace on a keyboard sends a command to the computer.

B. No — backspace corresponds to a control signal that affects the monitor.

C. Yes — the backspace sends the same character as the back arrow key.

**D. Yes — as far as the hardware is concerned, backspace is just another ascii character.**

9. What does the "read integer" service do?

A. It reads two's complement integers directly from the keyboard.

B. It reads a single digit from the keyboard and returns it in $v0.

C. It reads the next number on the current line. If there is no next number it goes on to the next line.

**D. It reads in a line of characters from the keyboard and then converts those characters to a two's complement integer.**

10. With a full-sized operating system, when a user enters a string of characters in response to a prompt, where are editing characters (like delete and backspace) interpretted?

A. In the user's program.

**B. An operating system service is running as the user enters and edits text.**

C. The keyboard device driver implements line editing.

D. The basic IO system is in charge of this.

1. What is the bitwise not of 0110 1010?

A. 0000 0000

B. 1111 1111

**C. 1001 0101**

D. 1010 1011

2. Translate the following pseudoinstruction into basic assembly language:

not $t5,$s1

A. not $t5,$s1,$0

B. not $t5,$s1

C. nor $t5,$s1

**D. nor $t5,$s1,$0**

3. Which of the following instructions will be translated by the extended assembler into one or more basic instructions?

A. or $s0,$t6,$t7

**B. or $s0,$t0,0xffff**

C. addiu $t0,$t5,32

D. subu $s0,$t3,$t6

4. Translate the following pseudoinstruction into basic assembly language:

negu $s1,$t1

A. addu $s1,$0,-$t1

B. nor $s1,$0,$t1

C. sub $s1,$t1,$0

**D. sub $s1,$0,$t1**

5. Write the pseudoinstruction that multiplies $t3 by $s0 and puts the result in $v0

**A. mul $v0,$s0,$t3**

B. mult $v0,$s0,$t3

C. mulu $v0,$s0,$t3

D. multi $v0,$s0,$t3

6. Translate the following pseudoinstuction into basic instructions:

div $s0,$t0,$t1

A. divu $s0,$t0,$t1

**B. div $t0,$t1**

**mflo $s0**

C. div $t0,$t1

mfhi $s0

D. div $t1,$t0

mflo $s0

7. Translate the following pseudoinstuction into basic instructions:

remu $s0,$t0,$t1

A. divu $s0,$t0,$t1

B. div $t0,$t1

mflo $s0

**C. div $t0,$t1**

**mfhi $s0**

D. div $t1,$t0

Mfl0o $s0

8. Rotate the following bit pattern two positions RIGHT:

10001101

A. 00110100

**B. 01100011**

C. 10001101

D. 10000001

9. Rotate the following bit pattern two positions LEFT:

10001101

**A. 00110110**

B. 00110100

C. 10001101

D. 10000001

10. Which sequence rotates the bits in $t0 two positions right?

**A.**

**li $t1,2**

**ror $t0,$t0,$t1**

B.

li $t1,-2

ror $t0,$t0,$t1

C.

rol $t0,$t0,-2

D.

rol $t0,$t0,2

1. Translate the following pseudoinstruction into basic instructions:

beqz $t0,spot

A. beqz $t0,$0,spot

B. beqz $t0,$t0,spot

C. beq $t0,spot

**D. beq $t0,$0,spot**

2. With what addresses may the b instruction be used with?

A. Any address in memory.

B. Any address in the first 64K of memory.

**C. Any address within 32K of the current address.**

D. Any address within the current program.

3. What format of data do the branch instructions assume?

A. unsigned integers only

B. two's complement integers only

C. Depending on the instruction, unsigned integers or two's complement integers.

**D. Depending on the instruction, unsigned integers, two's complement integers, or abstract bit patterns.**

4. Which of the following instruction sequences computes the absolute value of the integer in $t0? (Branch delays are turned off.)

A.

bgez $t0,pos

subi $t0,$0,$t0

pos:

**B.**

**bgez $t0,pos**

**subu $t0,$0,$t0**

**pos:**

C.

blez $t0,pos

subu $t0,$0,$t0

pos:

D.

bnez $t0,pos

subu $t0,$0,$t0

pos:

5. What register do some of the branch instructions use when they are translated into basic instructions?

A. $t0

B. $a0

**C. $at**

D. hi

6. Pick the syntactically incorrect instruction from the following:

A. bge $t1,$t3,lbl

B. bge $t1,-98,lbl

**C. bge $t1,value,lbl**

D. bge $t1,124034,lbl

7. Pick the syntactically incorrect instruction from the following:

A. slt $v0,$s1,123

B. slt $v0,$s1,$s3

C. slt $v0,$s1,0x123

**D. slt $t1,$s1,lbl**

8. Is it possible that a given pseudoinstruction could be translated into several different sequences of basic instructions?

A. No. There is only one possible translation of any given pseudoinstruction.

**B. Yes. Each pseudoinstruction asks for a little program of basic instructions, and many little programs do the same thing.**

9. Here is a fragment of code:

.data

array: .byte 12,23,45,12,-5,72

Which of the following pseudoinstructions loads register $s1 with the 45?

A.

li $t0,2

lw $s1,array($t0)

B.

li $t0,8

lw $s1,array($t0)

**C.**

**li $t0,2**

**lb $s1,array($t0)**

D.

li $t0,2

lw $s1,$t0(array)

10. Here is a fragment of code. (Note the subtle change from the previous question.)

.data

array: .word 12,23,45,12,-5,72

Which of the following pseudoinstructions loads register $s1 with the 45?

A.

li $t0,2

lw $s1,array($t0)

**B.**

**li $t0,8**

**lw $s1,array($t0)**

C.

li $t0,2

lb $s1,array($t0)

D.

li $t0,2

lw $s1,$t0(array)

11. Here is a fragment of code.

.data

string: .asciiz "Hello Virtual World!"

Which of the following pseudoinstructions loads register $s1 with the 'V'?

A.

li $t0,5

lw $s1,string($t0)

**B.**

**li $t0,6**

**lb $s1,string($t0)**

C.

li $t0,7

lb $s1,string($t0)

D.

li $t0,12

lw $s1,$t0(string)

12. Where do indexes begin for arrays in C or Java?

**A. 0**

B. 1

C. at any index the programmer picks

D. at any multiple of four

1. What phrase matches how a stack behaves?

A. First In First Out

B. First Out First In

**C. Last In First Out**

D. Last In Last Out

2. Adding an item to a stack is called:

A. pop

**B. push**

C. insert

D. add

3. Removing an item from the top of the stack is called:

**A. pop**

B. push

C. remove

D. delete

4. Assume that the stack pointer has already been set up. Which of the following code fragments pushes a word of data from register $t0?

A.

subu $sp,$sp,4

lw $t0,($sp)

B.

addu $sp,$sp,4

lw $t0,($sp)

C.

sw $t0,($sp)

subu $sp,$sp,4

**D.**

**subu $sp,$sp,4**

**sw $t0,($sp)**

5. As data is added to the MIPS stack, how does it grow?

A. It grows upward toward higher addresses.

**B. In grows downward toward lower addresses.**

C. It grows toward the address in the stack pointer.

D. It grows in whichever direction is available.

6. Assume that the stack pointer has already been set up and that the stack contains data. Which of the following code fragments POPS a word of data to register $t0?

A.

sw $t0,($sp)

addu $sp,$sp,4

B.

lw $t0,($sp)

subu $sp,$sp,4

**C.**

**lw $t0,($sp)**

**addu $sp,$sp,4**

D.

addu $sp,$sp,4

lw $t0,($sp)

7. How much total virtual memory is available with a MIPS processor?

A. 32 Megabytes

B. 512 Megabytes

C. 2 Gigabytes

**D. 4 Gigabytes**

8. Does the stack contain data that was declared in the .data section of a program?

**A. No. That goes in the data segment of memory.**

B. No. That goes in the text segment of memory.

C. Yes. The stack contains all the data of the program.

D. Yes. The .data section goes in the start of the stack.

9. What does the instruction lbu $t0,string do?

A. It copies a word of memory from symbolic address string into $t0.

B. It copies one byte of memory from symbolic address string into the low order byte $t0, leaving the other bytes of $t0 unchanged.

**C. It copies one byte of memory from symbolic address string into the low order byte $t0, and zeros the remaining bytes of $t0.**

D. It copies one byte of memory from symbolic address string into the low order byte $t0, and extends bit 7 into the remaining bytes of $t0.

10. Must a stack-based algorithm (such as string reversal) start out with an empty stack?

**A. No. The algorithm is not affected by data already on the stack.**

B. No. The algorithm will clear old data left on the stack.

C. Yes. If there is data already on the stack it will be inappropriately used.

D. Yes. Each program must have its own stack.

1. What is the major limitation in using a jump instruction to pass control to a subroutine?

A. The subroutine can not be passed any arguments.

B. The jump instruction is too slow for subroutine calls.

C. Subroutines are often distant in memory from the main routine, and the jump instruction can not reach them.

**D. The jump instruction gives the subroutine no information about how to return to the caller.**

2. A main routine passes control to a subroutine subA. Which of the following is usually true?

**A. When subA is done it returns control to main a few statements after where it was called.**

B. When subA is done it returns control to the the start of main.

C. When subA is done it returns control to the operating system.

D. When subA is done the whole program is finished.

3. What is a return address?

A. the address in the subroutine that gets control.

B. the address of the instruction that calls a subroutine.

**C. the address of the instruction in the caller to which the subroutine returns control**.

D. the address in the subroutine of the instruction that returns control to the caller.

4. Recall how the jal instruction works:

jal sub # $ra <— PC+4 $ra <— address 8 bytes away from the jal

# PC <— sub load the PC with the subroutine entry point

Say that the jal instruction is at address 0x400000. The subroutine sub is at address 0x400300.

What is in $ra after the jal instruction executes?

A. $ra == 0x400004

**B. $ra == 0x400008**

C. $ra == 0x400300

D. $ra == 0x400308

5. Is the jal instruction followed by a branch delay?

A. No.

**B. Yes.**

6. What does the following instruction do?

jr $s0

A. It immediately jumps to the address in $s0 with no branch delay.

B. It jumps to the address in $ra after a one instruction branch delay.

**C. It jumps to the address in $s0 after a one instruction branch delay.**

D. The instruction is illegal.

7. By software convention, which registers must a subroutine NOT change?

A. $t0 - $t9

**B. $s0 - $s7**

C. $a0 - $a3

D. $v0 - $v1

8. By software convention, which registers MAY a subroutine change?

**A. $t0 - $t9**

B. $s0 - $s7

C. $at - $gp

D. $k0 - $k1

9. What is the name for a symbol in a subroutine that is made visible to other routines?

A. local symbol

B. express symbol

**C. global symbol**

D. universal symbol

10. With the simple linkage convention, is it possible for main to call a subroutine which then calls another subroutine?

A. Yes. Subroutines will return to their caller in the opposite order they were called.

B. Yes. The jal and jr instructions automatically allow this.

**C. No. The first subroutine can't use a jal instruction without destroying the return address to main.**

D. No. This is never done in programming.

1. In the stack-based linkage convention, what subroutines must save the return address to their caller on the stack?

A. all subroutines

B. only main

**C. any subroutine that calls another subroutine.**

D. any subroutine that does not call another subroutine.

2. Which of the following code fragments saves the return address?

A. sub $ra,$ra,4

sw $sp,($ra)

**B. sub $sp,$sp,4**

**sw $ra,($sp)**

C. add $ra,$ra,4

sw $sp,($ra)

D. sw $ra,($sp)

sub $sp,$sp,4

3. When an operating system calls a main routine, what is in $ra?

A. $ra is all zeros to indicate the beginning of the call chain.

**B. The return address to a point in the operating system.**

C. The entry point to main

D. Random information.

4. Subroutine A calls subroutine B. Subroutine A has information in $t0 that it needs preserved. All the S registers are in use. What can be done?

A. Subroutine A saves $t0 in a V register.

B. Subroutine B saves $t0 on the stack.

C. Subroutine A saves $t0 on the stack after calling subroutine B.

**D. Subroutine A saves $t0 on the stack before calling subroutine B.**

5. What are the two jobs of a subroutine's prolog?

**A. Save the return address (if needed) and save any S registers it might use.**

B. Initialize variables and retrieve arguments.

C. Save the return address (if needed) and save any T registers it might use.

D. Put return values in V registers and restore S registers.

6. What are the jobs of a subroutine's epilog?

A. (i) put returned values in S registers, (ii) pop the return address, if needed, (iii) jr back to the caller.

B. (i) pop the return address, if needed, (ii) jr back to the caller.

**C. (i) put returned values in V registers, (ii) pop any S registers that were pushed, (iii) pop the return address, if needed, (iv) jr back to the caller.**

D. (i) put returned values in A registers, (ii) pop any T registers that were pushed, (iii) pop the return address, if needed, (iv) jr back to the caller.

7. In what order are registers popped off the stack when they are being restored?

A. The same order that they were pushed.

**B. The reverse of the order that they were pushed.**

C. Alphabetical order.

D. Numerical order.

8. Is there any limit to how deep subroutine calls can be nested?

A. Yes. Only main can call a subroutine, the the nesting is one level deep.

B. Yes. Calls can be nested no more than 8 levels deep.

**C. No, as long as there is memory for the stack, calls can be nested any level deep.**

D. No, as long as there is memory on the hard disk to give each subroutine a virtual address space.

9. During execution of an application made up of many subroutines, how does the activation chain behave?

**A. It repeatedly grows and shrinks depending on what subroutines are active.**

B. It constantly grows longer and longer.

C. It grows longer and develops branches as subroutines call different subroutines.

D. It grows to its maximum length and then stays there until the program is over.

10. What was the first prominent programming language that used stack-based subroutine linking?

A. FORTRAN

**B. Algol**

C. COBAL

D. C

1. In a high-level language, how is a local variable implemented?

A. As a section of global memory.

B. On the run-time heap.

C. As a slot in an array.

**D. As a location on a subroutine's stack-frame.**

2. What registers are the caller-saved registers?

**A. The T registers.**

B. The V registers.

C. The S registers.

D. The A registers.

3. In this calling convention, how many bytes are in each item on the stack?

A. 1

B. 2

**C. 4**

D. 8

4. What is the role of $fp with a stack frame?

A. $fp is used when values are popped and pushed during arithmetic evaluation.

**B. $fp points at the section of the stack that does not change as a subroutine executes.**

C. $fp is used to access values that the caller saved on the stack.

D. $fp points at global values all subroutines may access.

5. Does the number of registers that MIPS has limit the number of variables that a subroutine may have?

A. No, because each register may be used for any number of variables**.**

**B. No, because variables are implemented as locations on the stack.**

C. Yes, there may be only 32 variables in a program, minus the number of registers used for special purposes.

D. Yes, there may be only 8 variables in a program.

6. In the programming language C, what is the name for variables implemented as sections of the run-time stack?

**A. automatic**

B. dynamic

C. stack

D. heap

7. What is it called when a subroutine may call itself?

A. automatic

B. dynamic

C. static

**D. recursive**

8. When is space made on the stack for the local variables of a subroutine?

A. In the subroutine call of the caller.

**B. In the subroutine's prolog.**

C. In the subroutine's main body.

D. In the subroutine's epilog.

9. Say that a subroutine has just one variable. How is that variable copied into $t0 in the subroutine's body?

**A. lw $t0,0($fp)**

B. lw $t0,0($sp)

C. lw $t0,4($fp)

D. lw $t0,-4($fp)

10. Say that a subroutine has 3 local variables. How does it allocate space on the stack for these variables?

A. addu $fp,$sp,12

B. addu $fp,$sp,3

**C. subu $fp,$sp,12**

D. addu $sp,$fp,12

1. In the decimal fraction 12.345 what power of 10 is the digit 5 associated with?

**A. -3**

B. -2

C. -1

D. 1

2. In the binary fraction 10.001 what power of 2 is the rightmost bit associated with?

**A. -3**

B. -2

C. 1

D. -1

3. Express 1 + 4/10 + 3/100 using base 10 positional notation.

A. 1.043

B. 143.00

**C. 1.43**

D. 0.143

4. Express 1 + 1/2 + 0/4+ 1/8 using base 2 positional notation.

A. 1.11

B. 1.011

C. 11.1

**D. 1.101**

5. What is 2-1 ? (Express the answer in base 10)

**A. 1/2**

B. 1/4

C. 1/8

D. 2

6. Write 1.012 as a base 10 expression.

A. 1.125

**B. 1.25**

C. 1.5

D. 1.625

7. Say that the following is a four bit binary fixed-point expression and that the point is fixed between the middle two bits.

1011

What value does it represent? (Write the answer in decimal.)

A. 10.75

B. 2.3

**C. 2.75**

D. 1.625

8. Here is another four bit binary fixed-point expression with the point is fixed between the middle two bits.

0001

What value does it represent? (Write the answer in decimal.)

A. 1.25

B. 2.2

C. 1.625

**D. 0.25**

9. Perform the following addition of fixed-point binary operands using the Binary Addition Algorithm. Assume that the binary point is in the middle. What value does the sum represent?

1011

0001

----

A. 1100 represents 1210

B. 0110 represents 1.510

C. 1100 represents 3.510

**D. 1100 represents 310**

10. A method that uses only four bits can represent only 16 values. This is described as a limit on its \_\_\_\_\_\_\_\_.

A. Accuracy

**B. Precision**

C. Magnitude

D. Resolution

11. Which one of the following statments is true?

A. Computer arithmetic is so precise that if the input values are correct, all calculations will be accurate.

B. Computer arithmetic is always much more precise than an ordinary electronic calculator.

**C. Even when using double precision floating point it is easy for errors in a calculation to accumulate until the result is meaningless.**

D. Sixtyfour bits of precision is more than anyone could ever want.

12. The number 0.625 is here represented as a decimal fraction. Multiply it by two. Copy the one's place digit of the result to the beginning of a binary fraction. What is the beginning the binary fraction that represents the number?

**A. 0.1**

B. 0.0

C. 1.1

D. 2.0

13. After multiplying the number by two and dropping the digit to the right of the decimal point you are left with 0.25. Multiply this by two and copy the one's place digit of the result to the binary fraction. What is the binary fraction so far?

A. 0.11

**B. 0.10**

C. 1.01

D. 0.01

14. The number you are working on now looks like: 0.5. Repeat the above process to complete the binary fraction. The number 0.625 represented as a binary fraction is:

A. 0.111

B. 1.011

C. 0.110

**D. 0.101**

15. Represent the decimal fraction 0.375 as a binary fraction.

A. 0.11

B. 0.111

**C. 0.011**

D. 0.001

16. Which of the following binary fractions is an approximation to the decimal fraction 0.8?

**A. 0.11001**

B. 0.10101

C. 0.11010

D. 0.0111

17. If you know how many times a loop should execute, the loop control variable should be:

A. float

B. double

C. boolean

**D. int**

18. Say that you have a while loop that uses a conditional expression that involves floating point variables. Which one of the following comparison operators should not be used in the conditional expression?

**A. ==**

B. >

C. <

D. <=

19. Which one of the following decimal fractions cannot be represented accurately using a binary fraction?

A. 0.5

B. 0.0

**C. 0.1**

D. 0.125

20. Most programs that do floating point calculations should use variables of what type?

**A. double**

B. float

C. single

D. real

1. Express 5 + 0/2 + 1/4 + 0/8 + 1/16 using base 2 positional notation.

A. 5.0110

**B. 101.0101**

C. 101.101

D. 111.0101

2. Express 110.10012 as a sum of decimal integers and fractions

**A. 4 + 2 + 1/2 + 1/16**

B. 4 + 2 + 1/2 + 1/32

C. 4 + 2 + 1/4 + 1/16

D. 6 + 3/16

3. Express 1001.10112 as a sum of a single decimal integer and a single decimal fraction.

A. 10 + 11/2

B. 9 + 11/8

**C. 9 + 11/16**

D. This cannot be done.

4. Express the decimal value 7 3/8 in binary.

A. 101.11

B. 111.11

**C. 111.011**

D. 111.001

5. Express the decimal value 0.6250 as binary.

**A. 0.1010**

B. 0.0110

C. 0.1100

D. 0.1001

6. Express 0.3125 as binary

A. 0.1001

B. 0.1100

C. 0.1011

**D. 0.0101**

7. Express the binary 1.1001 as a decimal fraction.

**A. 1.5625**

B. 1.7500

C. 0.6252

D. 1.8750

8. Express the decimal value 0.73 as an approximation in binary

A. 0.11010

**B. 0.10111**

C. 0.01101

D. 0.10010

9. Express the decimal 0.5940 in a binary approximation.

A. 0.10010

B. 0.10101

C. 0.011111

**D. 0.10011**

10. Express the decimal 0.34375 in binary

**A. 0.01011**

B. 0.01101

C. 0.01010

D. 0.11011

1. What does the "IEEE" (as in "IEEE 754 Floating Point") stand for?

A. Industry Electrical Evaluation Enterprise

B. International Enterprise for Electronics Education

**C. Institute of Electrical and Electronics Engineers**

D. Interstate Engineering and Electrification Effort

2. How is 1.234 × 103 normally written?

A. .0001234

B. 12.34

C. 123.4

**D. 1234**

3. Write 0.0345 in scientific notation

**A. 3.45 × 10-2**

B. 3.45 × 10-3

C. 3.45 × 10+3

D. -3.45 × 102

4. Which part of 1.2345 × 1023 is the mantissa?

A. 23

B. .2345

C. 10

**D. 1.2345**

5. Write 0.00623 in scientific notation.

**A. 6.23 × 10-3**

B. .623 × 10-3

C. 6.23 × -103

D. 6.23 × -10-3

6. Which expression of the following is how 1.223 × 10-5 would appear in a source file.

A. 1.223e05

B. 1.223EE5

**C. 1.223e-5**

D. 1.223-5

E. 1.223E^5

7. A 32-bit IEEE 754 float consists of three fields: sign bit, biased exponent, and mantissa. How many bits are in each one?

A. 1, 12, 19

B. 2, 15, 15

**C. 1, 8, and 23**

D. 1, 4, 27

8. Here is a formula that says how the three fields combine to represent a number: value = (-1)s × 1.M × 2E-127

What biased exponent would be used to represent 22?

A. 310, represented as 0000 0011

B. -12410, represented as 1000 0100

C. 12810, represented as 1000 0000

**D. 12910, represented as 1000 0001**

9. Say that you wish to represent 001.00112 as a IEEE 754 float. What is the 23-bit mantissa?

A. 100110000000000000000000

B. 001100000000000000000000

**C. 00110000000000000000000**

D. 11000000000000000000000

10. Say that you wish to represent 010.10110112 as a IEEE 754 float. What is the 23-bit mantissa?

**A. 01011011000000000000000**

B. 101011011000000000000000

C. 0101101100000000000000

D. 01010110110000000000000

1. What advantage does 64-bit double precision have over 32-bit single precision?

A. Increased precision.

B. A greater range of values is covered.

C. Greater speed.

**D. Increased precision and a greater range of values.**

2. How many single precision floating point registers does MIPS have?

A. 8

B. 16

**C. 32**

D. 64

3. What is special about register $f0?

A. It always holds a floating point zero.

B. It always holds an integer zero.

C. It gets the results of a multiply operation.

**D. Nothing is special about it.**

4. With actual hardware is there a load delay associated with loading a floating point register from memory?

A. No. Only integers cause a load delay.

B. No. Load delays do not apply to coprocessor 0.

**C. Yes. To main memory a floating point representation is just a bit pattern like any other.**

D. Yes. Floating point bits take longer to move.

5. Does each possible 32-bit pattern represent a single precision float?

A. No. Most 32-bit patterns represent only integers.

**B. No. Some 32-bit patterns put illegal values in the fields of the float.**

C. Yes. Each possible 32-bit patterns represents a particular float.

D. Yes. But some numbers are represented by several 32-bit patterns.

6. Which of the following program fragments copies a floating point value from spotA to spotB?

**A.**

**l.s $f4,spotA**

**s.s $f4,spotB**

B.

s.s $f4,spotB

l.s $f4,spotA

C.

lf $f4,spotA

sf $f4,spotB

D.

li.f $f4,spotA

si.f $f4,spotB

7. Can a general purpose register be used with a floating point instruction?

A. Yes, as long as the bit pattern in it represents a float.

B. Yes, as long as at least one other operand is in a floating point register.

**C. No. It can hold a floating point representation, but can't be used in a floating point operation.**

D. No. General purpose registers can't hold floating point representations.

8. A single precision float has 24 bits of precision. This is equivalent to how many digits in a decimal representation?

**A. 7 or 8**

B. 9 or 10

C. 12

D. 24 or 25

9. Which of the following code fragments subtracts valB from valA, ensures that the result is positive, and stores the result in result?

A.

l.s $f6,valA

l.s $f8,valB

sub.s $f4,$f6,$f8

abs.s $f4

s.s $f4,result

**B.**

**l.s $f6,valA**

**l.s $f8,valB**

**sub.s $f4,$f6,$f8**

**abs.s $f4,$f4**

**s.s $f4,result**

C.

li.s $f6,valA

li.s $f8,valB

sub.s $f4,$f6,$f8

abs.s $f4,$f4

sf.s $f4,result

D.

l.s $f6,valB

l.s $f8,valA

sub.s $f4,$f6,$f8

abs.s $f4,$f4

s.s $f4,result

10. Here is a polynomial ax3 + bx2 + cx + d. Factor this polynomial to show how it would be evaluated using Horner's method.

**A. x( x( ax + b) + c) + d**

B. cx + d + ax3 + bx2

C. x3( a + bx-1) + cx-2) + d

D. (a + b + c + d)(x3 + x2 + x1 + 1)

1. How does a floating point comparison instruction indicate that a certain condition is true?

A. It sets the condition bit to zero**.**

**B. It sets the condition bit to one.**

C. It sets a designated register to one.

D. It executes a branch.

2. Why is it often unwise to test if two floating point values are equal?

**A. Floating point values are often approximations so equality sometimes does not occur where mathematically it should.**

B. Because there is more information in testing if one value is larger than another.

C. Because equality is not defined for floating point values.

D. Because testing for equality is time consuming.

3. After the condition bit is set (or cleared), how long does it retain its value?

A. Forever.

B. For one instruction cycle.

C. Until it is tested in a branch instruction.

**D. Until another comparison instruction changes it.**

4. Which set of instructions branches to label if the contents of $f0 are less than the contents of $f2?

A.

c.lt.s $f0, $f2

bc1f label

B.

c.lt.s $f2, $f0

bc1t label

**C.**

**c.lt.s $f0, $f2**

**bc1t label**

D.

c.le.s $f0, $f2

bc1t label

5. Which set of instructions branches to label if the contents of $f0 are greater or equal to the contents of $f2?

A.

c.ge.s $f0, $f2

bc1t label

B.

c.lt.s $f2, $f0

bc1t label

C.

c.eq.s $f0, $f2

bc1t label

c.gt.s $f0, $f2

bc1t label

**D.**

**c.le.s $f2, $f0**

**bc1t label**

6. Which of the following code segments branches to label if 0.0 < x?

A.

l.s $f6,x

li.s $f0,0.0

c.lt.s $f6,$f0

bc1t label

B.

l.s $f6,x

li.s $f6,0.0

c.lt.s $f0,$f6

bc1t label

**C.**

**l.s $f6,x**

**li.s $f0,0.0**

**c.lt.s $f0,$f6**

**bc1t label**

D.

li.s $f6,x

li.s $f0,0.0

c.lt.s $f0,$f6

bc1t label

7. Which of the following code segments branches to label if 0.0 < x < 10.0?

A.

l.s $f6,x

li.s $f0,0.0

c.lt.s $f6,$f0

bc1f fail

li.s $f0,10.0

c.lt.s $f6,$f0

bc1t label

fail: . . .

label: . . .

**B.**

**l.s $f6,x**

**li.s $f0,0.0**

**c.lt.s $f0,$f6**

**bc1f fail**

**li.s $f0,10.0**

**c.lt.s $f6,$f0**

**bc1t label**

**fail: . . .**

**label: . . .**

C.

l.s $f6,x

li.s $f0,0.0

c.lt.s $f0,$f6

bc1f fail

li.s $f0,10.0

c.lt.s $f0,$f6

bc1t label

fail: . . .

label: . . .

D.

l.s $f6,x

li.s $f0,0.0

c.le.s $f0,$f6

bc1f fail

li.s $f0,10.0

c.le.s $f6,$f0

bc1t label

fail: . . .

label: . . .

8. What is the method that is often used for computing square root?

**A. Newton's Method**

B. Cauchy's Method

C. Pascal's Method

D. Elimination Method

9. About how many decimal places of accuracy does single precision floating point have?

A. 3 or 4

B. 4 or 5

**C. 6 or 7**

D. 8 or 9

10. When a floating point value is being computed in a loop, how is the loop usually ended?

A. After a few iterations.

B. After very many iterations.

C. When the computed value exactly matches a specified condition.

**D. When the computed value falls into a specified range.**

1. How much of the 4 gigabytes of virtual memory does a user program run in?

A. none of it

B. about one quarter

**C. roughly half**

D. all of it

2. If the maximum address is 0xFFFFFFFF, what address is half of that?

A. 0x0000FFFF

B. 0x77777777

**C. 0x7FFFFFFF**

D. 0xFFFF0000

3. What is it called when memory for a program is allocated as the program is running?

A. static memory allocation

**B. dynamic memory allocation**

C. stack memory allocation

D. virtual memory allocation

4. What segment of memory is used for the data that is statically declared in a program?

**A. data segment**

B. static segment

C. stack segment

D. heap

5. Which of the following code fragments requests 16 bytes of memory?

**A.**

**li $a0, 16**

**li $v0, 9**

**syscall**

B.

li $a0, 16

li $v0, 9

C.

li $a1, 16

li $v1, 9

syscall

D.

li $a0, 9

li $v0, 16

syscall

6. With an actual operating system (not the simple trap handler of SPIM) is dynamic memory every returned to the system?

A. No. Memory can never be returned because that would leave inaccessible gaps.

B. No. Since there is an unlimited amount of virtual memory there is no reason to every give any back.

**C. Yes. After being returned, it might be used again to satisfy another request for memory.**

D. Yes. But once returned it is no longer available.

7. With an actual operating system, can the address of a block of dynamic memory be predicted before it is allocated?

A. No, because virtual memory addresses follow no order.

**B. No, because the operating system uses complex algorithms to find an available block.**

C. Yes, blocks are always allocated in sequential order.

D. Yes, because the program can request particular addresses.

8. Say that the base address of the following structure is in register $s8 and that the order of data in the structure has not been altered.

struct

{

int valA;

int valB;

int rate;

}

Which of the following code fragments loads $t0 with rate?

A. lw $s8,8($t0)

B. li $t0,8($s8)

C. lw $t0,($s8)

**D. lw $t0,8($s8)**

9. In a high level language like C is the actual byte-level layout of data in a struct exactly determined by its declaration?

**A. No. Compilers are free to rearrange the order of variables and to insert extra bytes into the data.**

B. No. The operating system may rearrange the order of the variables.

C. Yes. High level languages specify exactly how data is laid out at the byte level.

D. Yes. C, C++, and other languages lay out structures in exactly the same way.

10. The grouping of data that C calls a struct is called what in Pascal and COBOL?

A. object

B. entity

**C. record**

D. trace

1. Which was the first book written on data structures?

A. Programming Paradigms

B. Fundamentals of Data Structures

C. DOS for Dummies

**D. The Art of Computer Programming**

2. What is a data structure?

A. Any collection of data in main memory.

**B. A collection of data that includes the structural relationships betweeen the data.**

C. A collection of data and subroutines that operate on the data.

D. Any empty form that can be filled in with data.

3. In a linear data structure, each element has a (1)\_\_\_\_\_\_\_\_\_\_\_ and a (2)\_\_\_\_\_\_\_\_\_, except for the first element which has no (3)\_\_\_\_\_\_\_\_\_\_ and the last element which has no (4)\_\_\_\_\_\_\_\_\_\_\_\_.

**A. (1) predecessor, (2) successor, (3) predecessor, (4) successor.**

B. (1) predecessor, (2) successor, (3) successor, (4) predecessor.

C. (1) head, (2) tail, (3) tail, (4) head.

D. (1) data, (2) link, (3) data, (4) link.

4. May the data in an element of a data structure itself have structure?

A. No. Data must be the same size as the MIPS word size.

B. No. Data must be a primitive type such as integer or float.

**C. Yes. The internal structure of the data in a node may be arbitrarily complex.**

D. Yes. It can have the same structure as the outer structure of all the data.

5. What are the two sections of a node in a linked list?

A. The data and the counter.

B. The data and the null.

**C. The data and the link to the successor node.**

D. The data and the link to the predecessor node.

6. Examine this fragment:

????? $s0,head

. . .

.data

head: .word 7

Replace ???? with the instruction that will place the address of head into $s0.

A. lw

B. li

**C. la**

D. sw

7. Examine this fragment:

????? $s0,head

. . .

.data

head: .word 7

Replace ???? with the instruction that will place the contents of head into $s0.

**A. lw**

B. li

C. la

D. sw

8. Examine the following code fragment, which traverses a linked list starting at the head. Each node is 8 bytes long.

la $s1,head

loop: ???? $s1,exit

....

lw $s1,4($s1)

b loop

What instruction should replace ????

A. jr

B. lw

C. bnez

**D. beqz**

9. What is the usual value used for null with a data structure?

**A. 0x00000000**

B. 0x00

C. 0xFFFFFFFF

D. 0x11111111

10. Are the nodes of a linked list always arranged in in order in main memory?

**A. No. This might happen, but usually does not.**

B. No. The nodes are always contiguous, but not may not be in order.

C. Yes. In a linear data structure the data in memory is laid out in order.

D. Yes. In order to traverse the structure the data must be in order.

1. Here is a description of one node of a linked list (using the language C):

struct size

{

int width;

int height;

struct size \*next; // this means a pointer to

// the next node.

} ;

This describes a node that consists of two integers followed by an address.

Which of the following code fragments dynamically allocates enough memory for the node and puts its address in $s0 ?

**A.**

**li $v0,9**

**li $a0,12**

**syscall**

**move $s1,$v0**

B.

li $v0,12

li $a0,8

syscall

move $s1,$v0

C.

li $v0,12

li $a0,8

syscall

move $v0,$s1

D.

li $v0,12

li $a0,8

syscall

move $a0,$s1

2. Here (again) is a description of one node of a linked list:

struct size

{

int width;

int height;

struct size \*next; // this means a pointer to

// the next node.

} ;

Assume that the address of a node is in $v0. Which of the following puts a width of 12 and a height of 17 into the node?

A.

li $t0,17

sw $t0,0($v0)

li $t0,12

sw $t0,4($v0)

B.

li $t0,12

sw $t0,4($v0)

li $t0,17

sw $t0,0($v0)

**C.**

**li $t0,12**

**sw $t0,0($v0)**

**li $t0,17**

**sw $t0,4($v0)**

D.

li $t0,12

sw $v0,0($t0)

li $t0,17

sw $v0,4($t0)

3. Here is a description of one node of a linked list:

struct size

{

int width;

int height;

struct size \*next; // this means a pointer to

// the next node.

} ;

Assume that $s1 points at node 1 and that $s2 points at node 2. Which of the following makes node 2 the successor of node 1?

A. sw $s2,4($s1)

**B. sw $s2,8($s1)**

C. sw $s1,0($s2)

D. sw $s1,4($s2)

4. Here is a description of one node of a linked list:

struct size

{

int width;

int height;

struct size \*next; // this means a pointer to

// the next node.

} ;

The field head contains the address of a linked list of such nodes. Which of the following segments visits each node of the list?

A.

lw $s0,done

loop: beqz $s0,head

. . . # do something

lw $s0,4($s0)

b loop

done:

B.

lw $s0,head

loop: beqz $s0,done

. . . # do something

lw $s0,$s0

b loop

done:

**C.**

**lw $s0,head**

**loop: beqz $s0,done**

**. . . # do something**

**lw $s0,8($s0)**

**b loop**

**done:**

D.

lw $s0,head

loop: beqz $s0,done

. . . # do something

sw $s0,4($s0)

b loop

done:

5. Here is a description of one node of a linked list:

struct size

{

int width;

int height;

struct size \*next; // this means a pointer to

// the next node.

} ;

$s0 is pointing to the last node of a linked list. Which of the following code segments attaches one more node to the list and points $s0 to it?

**A.**

**li $v0,9**

**li $a0,12**

**syscall**

**sw $v0,8($s0)**

**move $s0,$v0**

B.

li $v0,9

li $a0,12

syscall

sw $v0,8($v0)

move $s0,$v0

C.

li $v0,9

li $a0,8

syscall

sw $v0,4($v0)

move $s0,$v0

D.

li $v0,12

li $a0,8

syscall

sw $v0,8($v0)

move $s0,$v0

1. What is the name for a list of entry points used to call subroutines?

A. call table

B. entry table

C. call list

**D. jump table**

2. Here is a list of five entry points.

jtab: .word sub0

.word sub1

.word sub2

.word sub3

.word sub4

Which of the following sequences calls sub3?

A.

lw $t1,jtab+3

jalr $t1

**B.**

**lw $t1,jtab+12**

**jalr $t1**

C.

jal jtab+3

D.

jal jtab+12

3. Where does the instruction jalr $t0 put the return address?

A. $t0

B. $sp

**C. $ra**

D. $jr

4. What is it called when the machine code for a subroutine is loaded only when a running program requires it?

A. run time loading

B. linking

C. library loading

**D. dynamic loading**

5. Does each software object of an executing program need a copy of its methods.

A. No--all objects of any type can use any method in the system.

**B. No--all objects of the same type can share the code for their methods.**

C. Yes--to enforce modularity each object has its own copy of each method.

D. Yes--in order to access the data of an object, the code must be part of it.

6. What is a DISadvantage when running programs share subroutines?

A. The subroutines need to be written only once.

B. The sizes of the programs is reduced.

**C. Calling a subroutine is more complicated.**

D. System resources managed by the subroutines can be effectively coordinated.

7. What are the characteristics of a software object?

**A. A software object has identity, state, and behavior.**

B. A software object is any block of main memory.

C. A software object has variables, values, and entry points.

D. A software object has data and methods to initialize the data.

8. Here is an object constructed in static memory:

.data

object: .word print # methods

.word read

.word clear

.word 0 # data

.word 7

Which of the following invokes the read method of the object? (Assume that the method expects the address of the object in $a0.)

A.

la $a0,object

lw $t0,0($a0)

jalr $t0

**B.**

**la $a0,object**

**lw $t0,4($a0)**

**jalr $t0**

C.

la $a0,object

lw $t0,1($a0)

jr $t0

D.

lw $a0,object

lw $t0,8($a0)

jr $t0

9. Here is an object constructed in static memory:

.data

object: .word print # methods

.word read

.word clear

.word 0 # data

.word 7

Which of the following implements the clear method of the object? The method clears the second word of data to zero. (Assume that the method expects the address of the object in $a0.)

A.

clear: sw $0,4($a0)

jr $ra

B.

clear: lw $t1,8($a0)

jalr $t1

jr $ra

C.

clear: lw $t1,8($a0)

sw $t1,16($a0)

jr $ra

**D.**

**clear: sw $0,16($a0)**

**jr $ra**

10. If a program is changed so that one new method is added to an object type, how much larger does each object of that type become?

A. Larger by as many bytes as it takes for the code of the new method.

B. 0 bytes

**C. 4 bytes**

D. This is impossible to estimate.