**Quiz**

Top of Form

**Question 1 (1 point)**

When retrieving a value from memory, the value goes immediately to \_\_\_\_\_.

Question 1 options:

|  |  |
| --- | --- |
|  | Control Unit |
|  | ALU |
|  | Accumulator A |
|  | CPU |

**Question 2 (1 point)**

ALU performs operations on one or two operands under the control of \_\_\_\_\_.

Question 2 options:

|  |  |
| --- | --- |
|  | the System Clock |
|  | the Microcontroller |
|  | the Control Unit |
|  | the Bus Controller |

**Question 3 (1 point)**

A Flow Chart contains implementation details.

Question 3 options:

|  |  |
| --- | --- |
|  | True |
|  | False |

**Question 4 (1 point)**

Some of the things that the Design Phase of an Assembly Language program defines is \_\_\_\_\_.

Question 4 options:

|  |  |
| --- | --- |
|  | the analysis of the problem |
|  | what has to be done |
|  | user requirements |
|  | where the data and program will reside in the memory map |

**Question 5 (1 point)**

As a course standard, program data will start at memory address \_\_\_\_\_.

Question 5 options:

|  |  |
| --- | --- |
|  | 1000 |
|  | $1000 |
|  | $2000 |
|  | 2000 |

**Question 6 (1 point)**

As a course standard, program instructions will start at memory address \_\_\_\_\_.

Question 6 options:

|  |  |
| --- | --- |
|  | 1000 |
|  | $1000 |
|  | $2000 |
|  | 2000 |

**Question 7 (1 point)**

The instruction used to Load A from memory address $1000 was \_\_\_\_\_.

Question 7 options:

|  |  |
| --- | --- |
|  | adda $1001 |
|  | ldda $1000 |
|  | deca |
|  | ldaa $1000 |

**Question 8 (1 point)**

The instruction used to add the contents of memory address $1001 to A was \_\_\_\_\_.

Question 8 options:

|  |  |
| --- | --- |
|  | ldda $1000 |
|  | adda $1001 |
|  | deca |
|  | adda $35 |

**Question 9 (1 point)**

The instruction used to store the contents of A to memory $1002 A was \_\_\_\_\_.

Question 9 options:

|  |  |
| --- | --- |
|  | deca |
|  | staa $1001 |
|  | ldda $1000 |
|  | staa $1002 |

**Question 10 (1 point)**

One of the valid formats for comments when using AsmIDE is \_\_\_\_\_.

Question 10 options:

|  |  |
| --- | --- |
|  | // This is a comment |
|  | /\*\* This is a comment \*/ |
|  | ; This is a comment |
|  | REM This is a comment |

**Question 11 (1 point)**

The Assembler Directive that specifief $1000 as the memory address where program data started was \_\_\_\_\_.

Question 11 options:

|  |  |
| --- | --- |
|  | db $1000 |
|  | orig $1000 |
|  | ds $1000 |
|  | org $1000 |

**Question 12 (1 point)**

Labels are symbols defining a memory address and are always placed in column 1.

Question 12 options:

|  |  |
| --- | --- |
|  | True |
|  | False |

**Question 13 (1 point)**

An example of the Assmbler Directive that is used to Define Byte, which is the value of a byte that will be placed at a given memory address is \_\_\_\_\_.

Question 13 options:

|  |  |
| --- | --- |
|  | db $25 |
|  | org $25 |
|  | ds $25 |
|  | db $1000 $25 |

**Question 14 (1 point)**

The Assembler Directive that defined storage for a byte whose value is changing or generally not known is \_\_\_\_\_.

Question 14 options:

|  |  |
| --- | --- |
|  | db |
|  | orig |
|  | ds |
|  | org |

**Question 15 (1 point)**

The memory address that instruction **adda** q uses to retrieve data from is \_\_\_\_\_.

Question 15 options:

|  |  |
| --- | --- |
|  | $1000 |
|  | $1001 |
|  | $1002 |
|  | unknown |

**Question 16 (1 point)**

The instruction used to decrement the value in Accumulator A by 1 was \_\_\_\_\_.

Question 16 options:

|  |  |
| --- | --- |
|  | ldaa 1 |
|  | adda 1 |
|  | deca |
|  | sba |

**Question 17 (1 point)**

The instruction used to store the value in Accumulator A at memory address $1002 was \_\_\_\_\_.

Question 17 options:

|  |  |
| --- | --- |
|  | ldaa $1002 |
|  | staa r |
|  | load r |
|  | stta r |

**Question 18 (1 point)**

The Program Counter (PC) always point to \_\_\_\_\_.

Question 18 options:

|  |  |
| --- | --- |
|  | the current instruction |
|  | the next instruction |
|  | the previous instruction |
|  | the next data memory address available for data storage |

**Question 19 (1 point)**

Single Stepping through a program in the simulator can be achieved by clicking on the \_\_\_\_\_ button.

Question 19 options:

|  |  |
| --- | --- |
|  | Go |
|  | Count |
|  | Step |
|  | To Return |

**Question 20 (1 point)**

The extension of the file that is loaded into the 68HCS12 Simulator is \_\_\_\_\_?

Question 20 options:

|  |  |
| --- | --- |
|  | .s19 |
|  | .asm |
|  | .lst |
|  | .sym |

**Question 21 (2 points)**

Question 21 options:

Convert Decimal -32 to its 8-bit Signed HEX and BINARY equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001

HEX



BINARY



**Question 22 (2 points)**

Question 22 options:

Convert Decimal -92 to its 8-bit Signed HEX and BINARY equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001   
HEX



BINARY



**Question 23 (2 points)**

Question 23 options:

Convert Decimal -82 to its 8-bit Signed HEX and BINARY equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001

HEX



BINARY



**Question 24 (1 point)**

 Convert Signed HEX number $AA to its Decimal equivalent.

Decimal

Bottom of Form

**Question 25 (1 point)**

Question 25 options:

Convert Signed HEX number $80 to its Decimal equivalent.  
Decimal

-128

**Question 26 (1 point)**

What is the Unary Branch Instruction for **Branch Always** ?

Question 26 options:

|  |  |
| --- | --- |
|  | BRA |
|  | BRN |
|  | BMI |
|  | BNE |
|  | BPL |
|  | BHI |
|  | BHS |
|  | BLO |
|  | BLS |
|  | BGE |
|  | BGT |
|  | BLE |
|  | BLT |

**Question 27 (1 point)**

What is the Unsigned Branch Instruction for **<=** ?

Question 27 options:

|  |  |
| --- | --- |
|  | BRA |
|  | BRN |
|  | BMI |
|  | BNE |
|  | BPL |
|  | BHI |
|  | BHS |
|  | BLO |
|  | BLS |
|  | BGE |
|  | BGT |
|  | BLE |
|  | BLT |

**Question 28 (1 point)**

What is the Unsigned Branch Instruction for **<** ?

Question 28 options:

|  |  |
| --- | --- |
|  | BRA |
|  | BRN |
|  | BMI |
|  | BNE |
|  | BPL |
|  | BHI |
|  | BHS |
|  | BLO |
|  | BLS |
|  | BGE |
|  | BGT |
|  | BLE |
|  | BLT |

**Question 29 (1 point)**

What is the Unsigned Branch Instruction for **>=** ?

Question 29 options:

|  |  |
| --- | --- |
|  | BRA |
|  | BRN |
|  | BMI |
|  | BNE |
|  | BPL |
|  | BHI |
|  | BHS |
|  | BLO |
|  | BLS |
|  | BGE |
|  | BGT |
|  | BLE |
|  | BLT |

**Question 30 (2 points)**

Question 30 options:

Convert Decimal 85 to its 8-bit Unsigned HEX and BINARY equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001   
HEX



BINARY



**Question 31 (2 points)**

Question 31 options:

Convert Decimal 192 to its 8-bit Unsigned HEX and BINARY equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001   
HEX



BINARY



**Question 32 (2 points)**

Convert Decimal 240 to its 8-bit Unsigned HEX and BINARY A10equivalents.   
Do not forget to use **$** or **%** directly in front of the entered values – e.g. **$**81 **%**10000001   
HEX



BINARY



**Question 33 (1 point)**

Question 33 options:

Convert Unsigned HEX number $80 to its Decimal equivalent.  
Decimal



**Question 34 (1 point)**

Convert Unsigned HEX number $AA to its Decimal equivalent.  
Decimal

