HONOR CODE

- I have not used any online resources during the exam.
- I have not obtained any help either from anyone in the class or outside when completing this exam.
- No sharing of notes/slides/textbook between students.
- NO SMARTPHONES.
- **CANVAS ANSWERS MAY BE LOCKED AFTER 1ST TRY.**

Questions Sheet.

Read all of the following information before starting the exam:

- For each question fill out the appropriate choice and write text on Canvas page. Also type clearly on in the exam on the appropriate text.
- For essay questions (non multiple choice). Write down the answer first and then 1/2 sentences with reasons.
- IF THE MULTIPLE CHOICE ANSWER IS WRONG WE WILL MARK THE ANSWER WRONG. IF THE MULTIPLE-CHOICE ANSWER IS CORRECT, WE WILL READ THE WRITTEN PORTION.
- Show all work, clearly and in order, if you want to get full credit.
- I reserve the right to take off points if I cannot see how you logically got to the answer (even if your final answer is correct).
- Circle or otherwise indicate your final answers.
- Please keep your written answers brief; be clear and to the point.
- I will take points off for rambling and for incorrect or irrelevant statements. This test has seven problem sets.
- You have to upload your worksheet/scrap to coursyst after you complete the exam. You have up to 11:59pm Aug 14th to do that.
HONOR CODE

Questions Sheet.

Section Virtual Memory 17 points. Canvas Q1-Q31
- Common questions. Canvas Q1-Q2
- For the virtual address 0x11a39 answer the following Canvas Q3-Q12. (0.5 pt each)
- For the virtual address 0xab6e7 answer the following. Canvas Q13-Q22 (0.5 pt each)
- c. For the following virtual address 0x02974 answer the following (Canvas: 23-32) (0.5 pt each)

B. Easy. RISCV Blackbox. [10 Points]
- 33. What is the value of the registers a0,a1, a2 on line 5: CHECK ? (show your working and explain your answer). Write down in hex [5]
- 34. Lets say message* is now a 2 character string "J-". The value of a2 on line 5:CHECK is 0. What is the mystery character "-". It has to be an ascii character between '0'--'9' (note not value 0-9). ? Write down the digit. [5]

C. Lets Cache I (10pts)
- 35. What is the number of tag bits in Cache-A and Cache-B ? (1pt)
- 36. What is the number of index bits in Cache-A and Cache-B ? (1pt)
- 37. What is the number of offset bits in Cache-A and Cache-B ? (1pt)
- 38. What is the hit rate for Cache-A and Cache-B for loop 1? What types of misses do we get? (2 pts)
- 39. What is the hit rate for Cache-A and Cache-B when you execute loop 2? (5 pts)

D. Lets Cache II (10pts)
- 40. What is the bits for virtual address? (1)
- 41. What is the bits for physical address? (1)
- 42. Which associativity below will maximize cache size while maintaining the same Tag:Index:Offset? (1)
- 43. Which block size below will maximize cache size while maintaining the same Tag:Index:Offset? (1)
- 44. Assuming associativity and block size from questions 40 and 41 what is the number of blocks? (1)
- 45. Now we’re working with a direct mapped cache with same TIO and block size as 41. What is miss rate for code below (3)
- 46. You add an L2 and shrink the L1. L1 hit latency is 3 cycles. L2 hit latency is 50 cycles. L2 hit rate is 90%. L1 hit rate is 25%. Memory is 100 cycles What is AMAT ? (2)

E. RISC-V Single Cycle Datapath 16 points.
- 47. What is the number of register that beqjalr needs to read and write in a single cycle ? (1)
- 48. What is the RegWEn signal ? (1)
- 49. What is the branch comparison signal we are interested in? (1)
- 50. Which fields are passed to the branch comparison ? (1)
- 51. What is the logic for beqjalr signal ? (1)
- 52. Consider the following modifications to the Reg[] register file. (2)
53. What are the inputs to the register file? (2)
54. What are the inputs to the ALU? (1)
55. What are the changes to mux-A? (2)
56. What are the changes to mux-B? (2)
57. For beqjalr instruction which signal does WBsel choose? (2)
58. What are the changes to the writeback stage? (2)

F. RISC-V Pipeline 12 points.
59. What hazards existing between instruction 1 and 2? (1)
60. What hazards existing between instruction 2 and 3? (1)
61. What hazards existing between instruction 3 and 4? (1)
62. What hazards existing between instruction 4 and 5? (1)
63. How many cycles does instruction 2 stall for? (2)
64. How many cycles does instruction 3 stall for? (2)
65. How many cycles does instruction 4 stall for? (2)
66. How many cycles does instruction 5 stall for? (2)

G. Lets RISC-V II 10 points
67. We are designing a new RISC-V instruction with 16 registers. Assuming that you use the extra bits to extend the immediate field, what is the range of half-word instructions that can be reached using a branch instruction in this new format? (2)

Refer code below for remaining questions
68. What is the PC address of instruction 4:beq (hex)?
69. What is the PC address of instruction 9:addi (hex)?
70. What is the PC address of instruction 11:ecall (hex)?
71. What is the PC address of instruction 13:mul (hex)?
72. What is the machine code for instruction at address 0x1C (hex)? (2)
73. After the first pass of the assembler instruction at 0x20 does not have the label resolved. (1)
74. After the first pass of the assembler instruction at 0x28 does not have the label resolved. (1)
Section Virtual Memory 17 points. Canvas Q1-Q31

Refer slide deck L21-VM-III Week 8 if you need to.

The chart below shows how memory accesses are treated in a system. The table below describes the parameters in the memory system.

Please use the data below to answer question groups Q1,Q2,Q3,Q4 on canvas.

CAUTION: When converting from binary to hex you can always pad the MSB e.g., 10 1010 (6 bit field) in hex is 0010 1010 (2 0s padded in MSB) is 0x2a.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical address bits</td>
<td>16</td>
</tr>
<tr>
<td>Size of page</td>
<td>256 bytes</td>
</tr>
<tr>
<td>Virtual address bits</td>
<td>20</td>
</tr>
<tr>
<td>TLB Sets</td>
<td>4</td>
</tr>
<tr>
<td>TLB Ways</td>
<td>2</td>
</tr>
<tr>
<td>TLB Size</td>
<td>8 entries</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache block</td>
<td>8 bytes</td>
</tr>
<tr>
<td>Cache size</td>
<td>64 bytes</td>
</tr>
<tr>
<td>Cache Sets</td>
<td>8</td>
</tr>
<tr>
<td>Cache Ways</td>
<td>1</td>
</tr>
</tbody>
</table>

- VPN - Virtual page number
- Index (Set index of cache or TLB)
- PPN - Physical page number
- INVALID. TLB entry is invalid
- TLB-T (TLB Tag)

### TLB

#### Way 0

<table>
<thead>
<tr>
<th>Index</th>
<th>TLB-T</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0x69]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>[0x396]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>[0x46]</td>
<td>eb</td>
</tr>
<tr>
<td>3</td>
<td>[0x29c]</td>
<td>4a</td>
</tr>
</tbody>
</table>

#### Way 1

<table>
<thead>
<tr>
<th>Index</th>
<th>TLB-T</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0x3b]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>[0x010]</td>
<td>88</td>
</tr>
<tr>
<td>2</td>
<td>[0x2ad]</td>
<td>4b</td>
</tr>
<tr>
<td>3</td>
<td>[0x6]</td>
<td>dc</td>
</tr>
</tbody>
</table>
### Page Table (Partial)

**CAUTION:** Only partial table relevant to the questions are shown.

<table>
<thead>
<tr>
<th>VPN</th>
<th>PPN</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1a4</td>
<td>---</td>
<td>0</td>
</tr>
<tr>
<td>0xe59</td>
<td>---</td>
<td>0</td>
</tr>
<tr>
<td>0x11a</td>
<td>0xeb</td>
<td>1</td>
</tr>
<tr>
<td>0xa73</td>
<td>0x4a</td>
<td>1</td>
</tr>
<tr>
<td>0xec</td>
<td>0x00</td>
<td>1</td>
</tr>
<tr>
<td>0x29</td>
<td>0x12</td>
<td>1</td>
</tr>
<tr>
<td>0xab6</td>
<td>0x4b</td>
<td>1</td>
</tr>
<tr>
<td>0x1b</td>
<td>0xdc</td>
<td>1</td>
</tr>
<tr>
<td>0x8ff</td>
<td>0x91</td>
<td>1</td>
</tr>
<tr>
<td>0xbf1</td>
<td>0x47</td>
<td>1</td>
</tr>
<tr>
<td>0x016</td>
<td>0x99</td>
<td>1</td>
</tr>
<tr>
<td>0x010</td>
<td>0x88</td>
<td>1</td>
</tr>
</tbody>
</table>
Cache

- Way 0

<table>
<thead>
<tr>
<th>Index</th>
<th>Tag</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0x3ae]</td>
<td>0x39</td>
<td>0x24</td>
<td>0x76</td>
<td>0x62</td>
<td>0x17</td>
<td>0x92</td>
<td>0x72</td>
<td>0x38</td>
</tr>
<tr>
<td>1</td>
<td>[0x373]</td>
<td>0x24</td>
<td>0x6c</td>
<td>0x38</td>
<td>0x47</td>
<td>0x8e</td>
<td>0x43</td>
<td>0xc2</td>
<td>0x29</td>
</tr>
<tr>
<td>2</td>
<td>[0x2c7]</td>
<td>0x08</td>
<td>0x9d</td>
<td>0xd1</td>
<td>0xf8</td>
<td>0x53</td>
<td>0x05</td>
<td>0x14</td>
<td>0x16</td>
</tr>
<tr>
<td>3</td>
<td>[0xdc]</td>
<td>0xfa</td>
<td>0x8c</td>
<td>0x3d</td>
<td>0x11</td>
<td>0x14</td>
<td>0xf4</td>
<td>0xe9</td>
<td>0xc3</td>
</tr>
<tr>
<td>4</td>
<td>[0x12b]</td>
<td>0x11</td>
<td>0xfa</td>
<td>0x4c</td>
<td>0x9d</td>
<td>0xb3</td>
<td>0x1d</td>
<td>0xb6</td>
<td>0x87</td>
</tr>
<tr>
<td>5</td>
<td>[0x27e]</td>
<td>0xdc</td>
<td>0x0f</td>
<td>0x44</td>
<td>0x85</td>
<td>0x6b</td>
<td>0xfc</td>
<td>0x04</td>
<td>0x4f</td>
</tr>
<tr>
<td>6</td>
<td>[0x49]</td>
<td>0x38</td>
<td>0xde</td>
<td>0xfb</td>
<td>0x10</td>
<td>0x3d</td>
<td>0xfe</td>
<td>0x05</td>
<td>0xc5</td>
</tr>
<tr>
<td>7</td>
<td>[0x3ac]</td>
<td>0x49</td>
<td>0x52</td>
<td>0xe0</td>
<td>0xdd</td>
<td>0xf0</td>
<td>0x0d</td>
<td>0xa6</td>
<td>0xe0</td>
</tr>
</tbody>
</table>
Common questions. Canvas Q1-Q2

1. How many bits is the VPN (1pt)?
2. How many bits is the PPN (1pt)?

For the virtual address 0x11a39 answer the following Canvas Q3-Q12. (0.5 pt each)

- What is the VPN
- What is the TLB tag.
- Is it a TLB hit or miss
- Is it a page fault
- What is the PPN?
- What is the cache tag?
- What is the cache index
- What is the byte offset
- Is it a cache hit or miss
- What is the data byte

For the virtual address 0xab6e7 answer the following. Canvas Q13-Q22 (0.5 pt each)

- What is the VPN
- What is the TLB tag.
- Is it a TLB hit or miss
- Is it a page fault
- What is the PPN?
- What is the cache tag?
- What is the cache index
- What is the byte offset
- Is it a cache hit or miss
- What is the data byte

c. For the following virtual address 0x02974 answer the following (Canvas: 23-32) (0.5 pt each)

- What is the VPN
- What is the TLB tag.
- Is it a TLB hit or miss
- Is it a page fault
• What is the PPN?
• What is the cache tag?
• What is the cache index
• What is the byte offset
• Is it a cache hit or miss
• What is the data byte
B. Easy. RISC-V Blackbox. [10 Points]

WARNING: FOR THIS QUESTION MAKE SURE YOU WRITE YOUR CORRECT ANSWER AT THE BEGINNING. ALSO WRITE ONE/TWO SENTENCES REASONING YOUR ANSWER. OTHERWISE WE WILL SIMPLY ZERO IT OUT.

Assume we have two arrays input and output.

```
1 | char *message = "MESSAGE"
```

Study the following RISC-V code shown below and answer the questions. You can assume the data segments starts at 0x10000000 and only contains message. You can assume a0 contains value of message* at the start.

```
1 | .text
2 | main:
3 |   li  a2,0xFF
4 |   jal BLACKBOX
5 |   # CHECK
6 |   li a0,10
7 |   ecall
8 | BLACKBOX:
9 |   addi sp,sp,-4
10 |   sw ra, 0(sp)
11 |   lbu  a1,0(a0)
12 |   beq a1,zero,end
13 |   and a2,a1,a2
14 |   addi a0,a0,1
15 |   jal process
16 | end:
17 |   lw ra, 0 (sp)
18 |   addi sp,sp,4
19 |   ret
```

33. What is the value of the registers a0,a1, a2 on line 5: CHECK ? (show your working and explain your answer). Write down in hex [5]

34. Lets say message* is now a 2 character string "J-". The value of a2 on line 5:CHECK is 0. What is the mystery character "-". It has to be an ascii character between '0'--'9' (note not value 0-9). ? Write down the digit. [5]

C. Lets Cache I (10pts)
Assume we are working in a 4GB physical address space.

<table>
<thead>
<tr>
<th>Cache</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-A</td>
<td>Direct-mapped, 4KB, 512 byte blocks</td>
</tr>
<tr>
<td>Cache-B</td>
<td>Set-associate, 4KB, 2 ways, 512 byte block</td>
</tr>
</tbody>
</table>

Both caches use write-back and write-allocate policies.

Answer questions below

```c
int size = 2 * 1024;
// long long int is 8 bytes
long long int array[size];
/* loop 1 */
for (int i = 0; i < size; i++) {
    array[i] = i;
}

/* loop 2 */
for (int i = 0; i < size; i += 8) {
    // Order in which access happens
    // 3rd 1st 2nd
    array[i] = array[i] * arr[0];
}
```

35. What is the number of tag bits in Cache-A and Cache-B? (1pt)

36. What is the number of index bits in Cache-A and Cache-B? (1pt)

37. What is the number of offset bits in Cache-A and Cache-B? (1pt)

38. What is the hit rate for Cache-A and Cache-B for loop 1? What types of misses do we get? (2 pts)

39. What is the hit rate for Cache-A and Cache-B when you execute loop 2? (5 pts)

D. Lets Cache II (10pts)

We have a mystery cache. A mystery, byte addressed cache has Tag:Index:Offset (T:I:O) = 12:4:3.

40. What is the bits for virtual address? (1)

41. What is the bits for physical address? (1)
42. Which associativity below will maximize cache size while maintaining the same Tag:Index:Offset? (1)

43. Which block size below will maximize cache size while maintaining the same Tag:Index:Offset? (1)

44. Assuming associativity and block size from questions 40 and 41 what is the number of blocks? (1)

45. Now we’re working with a direct mapped cache with same TIO and block size as 41. What is miss rate for code below (3)

```c
#define LEN 2048
int ARRAY[LEN];
int main() {
    for (int i = 0; i < LEN - 32; i+=32) {
        ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+32];
        ARRAY[i] += 10;
    }
}
```

46. You add an L2 and shrink the L1. L1 hit latency is 3 cycles. L2 hit latency is 50 cycles. L2 hit rate is 90%. L1 hit rate is 25%. Memory is 100 cycles What is AMAT ? (2)

**E. RISC-V Single Cycle Datapath 16 points.**

We wish to introduce a new instruction into our RISC-V datapath. 
beqjalr. This instruction is a conditional jal instruction. Typical branch instruction has limited offsets since all offsets are PC-relative. Jalr however can specify register relative. The instruction works as follows. The instruction is encoded as follows.

- We no longer use an immediate for the branch offsets
- We revert imm[4:1|11] to specify a register

<table>
<thead>
<tr>
<th>original beq instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>imm[12] 10:5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>new beqjalr instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>f7 (0x40)</td>
</tr>
</tbody>
</table>
It combines the semantics of branch, JALR and R-type instruction.

- Like a branch it performs a comparison.
- If comparison succeeds it performs a JALR, the destination is contained in the rd register.
  rd is treated as a source.
- If comparison fails, it simply zeros out the register which contains the PC it should jump to.
  rd is treated as a destination.
- We have a new control signal beqjalr which is 1 if the instruction being decoded is a beqjalr.

Given the single cycle datapath below, select the correct modifications in parts such that the datapath executes correctly for this new instruction (and all other instructions!). You can make the following assumptions:

**Caution 2:** Pay careful attention to which input line is 1 and which line is 0 in the muxes. Some muxes choose top-most input as 0, some choose bottom-most input as 0.

**Hint:** YOU DO NOT REQUIRE TRUTH TABLES

Try writing down in plain english or reading out the logic to yourself e.g, !(A<=B) is A is not equal to B and A is not LT (less than) B.

```c
1  if (R[rs1] == R[rs2]) {
2    R[ra] = PC + 4;
3    PC = R[rd];
4  } else {
5    // Zero out the rd register and destroy
6    R[rd] = 0
7  }
```
Baseline Pipeline

Pipeline with beqjalr (Red boxes indicate questions)
47. What is the number of register that beqjalr needs to read and write in a single cycle? (1)

48. What is the RegWEn signal? (1)

49. What is the branch comparison signal we are interested in? (1)

50. Which fields are passed to the branch comparison? (1)

51. What is the logic for beqjalr signal? (1)

52. Consider the following modifications to the Reg[] register file. (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?

EQ is 1 if R[rs1] == R[rs2]. 0 otherwise
53. What are the inputs to the register file? (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?
54. What are the inputs to the ALU? (1)

55. What are the changes to mux-A? (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?
56. What are the changes to mux-B ? (2)
Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?

57. For beqjalr instruction which signal does WBsel choose ? (2)
Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?

58. What are the changes to the writeback stage ? (2)

F. RISC-V Pipeline 12 points.
Refer slide deck L29–Hazard Week 11 if you need to.

Consider a typical 5-stage (Fetch, Decode, Execute, Memory, WriteBack) pipeline. Assume pipeline registers exist where the dotted lines are.

This pipeline is more simple than the one you dealt with in the assignment. READ RULES BELOW

- Forwarding/Bypassing is not implemented; dependent instructions will have to wait in the IF stage.
- Following a branch, the next instructions stalls until branch is resolved.
- We can only read or write the register in a cycle.
- A stall is the number of cycles after the previous instruction that the current instruction can start.

We modify this pipeline to add another memory stage and create a six-stage pipeline.
Answer questions based on the following program

For the given code, what hazards might exist and due to which lines? Assume all registers have been initialised and that all labels are defined and that all branches are taken.

1. \texttt{bneq \textbf{a0}, zero, end}
2. \texttt{addi t0, t0, 1}
3. \texttt{lw s0, 0x10(t0)}
4. \texttt{mul s1, s0, \textbf{a0}}
5. \texttt{add s1, s0, \textbf{a0}}

59. What hazards existing between instruction 1 and 2 ? (1)
60. What hazards existing between instruction 2 and 3 ? (1)
61. What hazards existing between instruction 3 and 4 ? (1)
62. What hazards existing between instruction 4 and 5 ? (1)
63. How many cycles does instruction 2 stall for ? (2)
64. How many cycles does instruction 3 stall for ? (2)
65. How many cycles does instruction 4 stall for ? (2)
66. How many cycles does instruction 5 stall for ? (2)
G. Lets RISC-V II 10 points

67. We are designing a new RISC-V instruction with 16 registers. Assuming that you use the extra bits to extend the immediate field, what is the range of half-word instructions that can be reached using a branch instruction in this new format? (2)

Refer code below for remaining questions

```
1 | add s0, x0, x0
2 | addi s1, x0, 4
3 | loop:
4 |  beq s0, s1, end
5 |  add a0, a0, s0
6 |  jal ra, square
7 |  jal ra, printf
8 | n:
9 |   addi s0, s0, 1
10 |   j loop
11 | end:
12 |   ecall
13 | square:
14 |   mul a0, a0, a0
15 |   ret
```

68. What is the PC address of instruction 4:beq (hex) ?

69. What is the PC address of instruction 9:addi (hex) ?

70. What is the PC address of instruction 11:ecall (hex) ?

71. What is the PC address of instruction 13:mul (hex) ?

72. What is the machine code for instruction at address 0x1C (hex) ? (2)

73. After the first pass of the assembler instruction at 0x20 does not have the label resolved. (1)

74. After the first pass of the assembler instruction at 0x28 does not have the label resolved. (1)
<table>
<thead>
<tr>
<th>dec</th>
<th>oct</th>
<th>hex</th>
<th>ch</th>
<th>dec</th>
<th>oct</th>
<th>hex</th>
<th>ch</th>
<th>dec</th>
<th>oct</th>
<th>hex</th>
<th>ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>40</td>
<td>20</td>
<td>(space)</td>
<td>64</td>
<td>100</td>
<td>40</td>
<td>@</td>
<td>96</td>
<td>140</td>
<td>60</td>
<td>`</td>
</tr>
<tr>
<td>33</td>
<td>41</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>101</td>
<td>41</td>
<td>A</td>
<td>97</td>
<td>141</td>
<td>61</td>
<td>a</td>
</tr>
<tr>
<td>34</td>
<td>42</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>102</td>
<td>42</td>
<td>B</td>
<td>98</td>
<td>142</td>
<td>62</td>
<td>b</td>
</tr>
<tr>
<td>35</td>
<td>43</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>103</td>
<td>43</td>
<td>C</td>
<td>99</td>
<td>143</td>
<td>63</td>
<td>c</td>
</tr>
<tr>
<td>36</td>
<td>44</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>104</td>
<td>44</td>
<td>D</td>
<td>100</td>
<td>144</td>
<td>64</td>
<td>d</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>105</td>
<td>45</td>
<td>E</td>
<td>101</td>
<td>145</td>
<td>65</td>
<td>e</td>
</tr>
<tr>
<td>38</td>
<td>46</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>106</td>
<td>46</td>
<td>F</td>
<td>102</td>
<td>146</td>
<td>66</td>
<td>f</td>
</tr>
<tr>
<td>39</td>
<td>47</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>107</td>
<td>47</td>
<td>G</td>
<td>103</td>
<td>147</td>
<td>67</td>
<td>g</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>110</td>
<td>48</td>
<td>H</td>
<td>104</td>
<td>150</td>
<td>68</td>
<td>h</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>111</td>
<td>49</td>
<td>I</td>
<td>105</td>
<td>151</td>
<td>69</td>
<td>i</td>
</tr>
<tr>
<td>42</td>
<td>52</td>
<td>2a</td>
<td>*</td>
<td>74</td>
<td>112</td>
<td>4a</td>
<td>J</td>
<td>106</td>
<td>152</td>
<td>6a</td>
<td>j</td>
</tr>
<tr>
<td>43</td>
<td>53</td>
<td>2b</td>
<td>+</td>
<td>75</td>
<td>113</td>
<td>4b</td>
<td>K</td>
<td>107</td>
<td>153</td>
<td>6b</td>
<td>k</td>
</tr>
<tr>
<td>44</td>
<td>54</td>
<td>2c</td>
<td>,</td>
<td>76</td>
<td>114</td>
<td>4c</td>
<td>L</td>
<td>108</td>
<td>154</td>
<td>6c</td>
<td>l</td>
</tr>
<tr>
<td>45</td>
<td>55</td>
<td>2d</td>
<td>-</td>
<td>77</td>
<td>115</td>
<td>4d</td>
<td>M</td>
<td>109</td>
<td>155</td>
<td>6d</td>
<td>m</td>
</tr>
<tr>
<td>46</td>
<td>56</td>
<td>2e</td>
<td>.</td>
<td>78</td>
<td>116</td>
<td>4e</td>
<td>N</td>
<td>110</td>
<td>156</td>
<td>6e</td>
<td>n</td>
</tr>
<tr>
<td>47</td>
<td>57</td>
<td>2f</td>
<td>/</td>
<td>79</td>
<td>117</td>
<td>4f</td>
<td>O</td>
<td>111</td>
<td>157</td>
<td>6f</td>
<td>o</td>
</tr>
<tr>
<td>48</td>
<td>60</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>120</td>
<td>50</td>
<td>P</td>
<td>112</td>
<td>160</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>49</td>
<td>61</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>121</td>
<td>51</td>
<td>Q</td>
<td>113</td>
<td>161</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>50</td>
<td>62</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>122</td>
<td>52</td>
<td>R</td>
<td>114</td>
<td>162</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>51</td>
<td>63</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>123</td>
<td>53</td>
<td>S</td>
<td>115</td>
<td>163</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>52</td>
<td>64</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>124</td>
<td>54</td>
<td>T</td>
<td>116</td>
<td>164</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>53</td>
<td>65</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>125</td>
<td>55</td>
<td>U</td>
<td>117</td>
<td>165</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>54</td>
<td>66</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>126</td>
<td>56</td>
<td>V</td>
<td>118</td>
<td>166</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>55</td>
<td>67</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>127</td>
<td>57</td>
<td>W</td>
<td>119</td>
<td>167</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>56</td>
<td>70</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>130</td>
<td>58</td>
<td>X</td>
<td>120</td>
<td>170</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>57</td>
<td>71</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>131</td>
<td>59</td>
<td>Y</td>
<td>121</td>
<td>171</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>58</td>
<td>72</td>
<td>3a</td>
<td>0</td>
<td>90</td>
<td>132</td>
<td>5a</td>
<td>z</td>
<td>122</td>
<td>172</td>
<td>7a</td>
<td>-</td>
</tr>
</tbody>
</table>