Jump Table

1.

```
      .quad  .L4
      .quad  .L6
      .quad  .L6
      .quad  .L7
      .quad  .L3
      .quad  .L8
      .quad  .L7
```

2.

```
[1]31/0x1f                        [2]6
[3]ja                             [4].L5(,%rax,8)
[5]movb %dl, (%rax)               [6]6
[7]movw 0x201f, (%rax)            [8]7
```

3.

```
0     1     2     3     4     5     6     7
0xf0  0x1f  0xf0  0xff  0x22  0x1f  0xf0  0x00
```

1. Jump table under label `.L5` is hidden. Please fill in the hidden part according to the C code given above.

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

Base: 0x1f

Length: 7 (0x1f – 0x25)

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

case 0x1f <-> .L4

```
.L4:
    cmpl     $0, -4(%rbp)
    jle     .L6
    movl      -4(%rbp), %eax
    movslq    %eax, %rdx
    movq      -24(%rbp), %rax
    addq     %rdx, %rax
    movl      -4(%rbp), %edx
    movslq    %edx, %rcx
    movq      -24(%rbp), %rdx
    addq     %rcx, %rdx
    movzbl    (%rdx), %edx
    movl     %edx, %esi
    movl     -4(%rbp), %edx
    movslq    %edx, %rdx
    leaq     -1(%rdx), %rcx
    movq     -24(%rbp), %rdx
    addq     %rcx, %rdx
    movzbl    (%rdx), %edx
    addl     %esi, %edx
          ________[5]________
```

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

```
              __________    ___________
.L6:
        movl     -4(%rbp), %eax
        movslq    %eax, %rdx
        movq     -24(%rbp), %rax
        addq     %rdx, %rax
        movl     -4(%rbp), %edx
        movslq    %edx, %rcx
        movq     -24(%rbp), %rdx
        addq     %rcx, %rdx
        movzbl    (%rdx), %edx
        orl      $-16, %edx
        movb     %dl, (%rax)
        jmp   .L9
```

case 0x20: case 0x21: <-> .L6

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

case 0x22: case 0x25: <-> .L7

```
.L7:
        movl      -4(%rbp), %eax
        leal      3(%rax), %edx
        testl      %eax, %eax
        cmovs      %edx, %eax
        sarl      $2, %eax
        leaq      0(,%rax,4), %rdx
        movq      -24(%rbp), %rax
        addq      %rax, %rdx
        movl      -4(%rbp), %eax
        movslq     %eax, %rcx
        movq      -24(%rbp), %rax
        addq      %rcx, %rax
        movzbl     (%rax), %eax
        movsbl     %al, %eax
        movl      %eax, (%rdx)
        jmp  .L9
```

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

```
.L8:
       movl     -4(%rbp), %eax
       movslq    %eax, %rdx
       movq     -24(%rbp), %rax
       addq     %rax, %rdx
       movq     -24(%rbp), %rax
       movq     (%rax), %rax
       sarq     $8, %rax
       movb     %al, (%rdx)
       jmp  .L9
```

case 0x24: <-> .L8

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

```
.L3:
        cmpl      __[6]__, -4(%rbp)
        jg    .L9
        movl      -4(%rbp), %eax
        movslq      %eax, %rdx
        movq      -24(%rbp), %rax
        addq      %rdx, %rax
        ________[7]________
```

default: <-> .L3

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {          .L4
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:     .L6
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];     .L7
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);     .L8
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }     .L3
      break;
    }
  }
}
```

```
.L5:
    .quad  .L4
    .quad  .L6
    .quad  .L6
    .quad  .L7
    .quad  .L3
    .quad  .L8
    .quad  .L7
```

```
movl        -4(%rbp), %eax
movslq       %eax, %rdx
movq        -24(%rbp), %rax
addq        %rdx, %rax
movzbl       (%rax), %eax
movsbl       %al, %eax
subl  Base: 0x1f[1]__, %eax
cmpl  length-1: 6[2]__, %eax
ja [3]__      .L3
movl        %eax, %eax
movq  .L5(,%rax,8)[4]__, %rax
jmp  *%rax
```

```
.L5:
    .quad   .L4
    .quad   .L6
    .quad   .L6
    .quad   .L7
    .quad   .L3
    .quad   .L8
    .quad   .L7
```

```

#define BUF_SIZE 8
void transfer(char *buf) {
    int i;
    for (i = 0; i < BUF_SIZE; i++) {
    switch (buf[i]) {
      case 0x1f:
```

```
movl      -4(%rbp), %eax
movslq     %eax, %rdx
movq      -24(%rbp), %rax
addq      %rdx, %rax
movzbl     (%rax), %eax
movsbl     %al, %eax
subl  Base: 0x1f[1]__, %eax
cmpl  length-1: 6[2]__, %eax
ja [3]__    .L3
movl      %eax, %eax
movq  .L5(,%rax,8)[4]__, %rax
jmp  *%rax
```

```
.L5:
    .quad  .L4
    .quad  .L6
    .quad  .L6
    .quad  .L7
    .quad  .L3
    .quad  .L8
    .quad  .L7
```

Other answers?

- [2] 7, [3] jae
- [1] 0x1f + X, [2] 6 + X, [4] .L5(-X, %rax, 8)

```
.L4:
        cmpl      $0, -4(%rbp)
        jle    .L6
        movl      -4(%rbp), %eax
        movslq    %eax, %rdx
        movq      -24(%rbp), %rax
        addq     %rdx, %rax
        movl      -4(%rbp), %edx
        movslq    %edx, %rcx
        movq      -24(%rbp), %rdx
        addq     %rcx, %rdx
        movzbl    (%rdx), %edx
        movl     %edx, %esi
        movl     -4(%rbp), %edx
        movslq    %edx, %rdx
        leaq     -1(%rdx), %rcx
        movq     -24(%rbp), %rdx
        addq     %rcx, %rdx
        movzbl    (%rdx), %edx
        addl     %esi, %edx
        _______[5] movb %dl, (%rax)________
```

buf[i] += buf[i-1]

- Type: char
- movb

```
.L3:
        cmpl      __6__[6]___,  -4(%rbp)
        jg        .L9
        movl      -4(%rbp),  %eax
        movslq      %eax, %rdx
        movq      -24(%rbp),  %rax
        addq      %rdx, %rax
        _________[7] movw $0x201f, (%rax)
```

```
If (i < BUF_SIZE -1) {
    *(short *)buf[i] = 0x201f;
}
```

- Type: short
- movw

```
.L2:
        cmpl    __7[8]__, -4(%rbp)
        jle     .L11
        popq    %rbp
        ret
```

```
for (i = 0; i < BUF_SIZE; i++) {

}
```

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

0	1	2	3	4	5	6	7
0x24	0xf0	0x22	0x1f	0x22	0x21	0x1f	0x21

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 0

0	1	2	3	4	5	6	7
0x24	0xf0	0x22	0x1f	0x22	0x21	0x1f	0x21
0xf0	0xf0	0x22	0x1f	0x22	0x21	0x1f	0x21

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 1

0	1	2	3	4	5	6	7
0x24	0xf0	0x22	0x1f	0x22	0x21	0x1f	0x21
0xf0	0x1f	0x20	0x1f	0x22	0x21	0x1f	0x21

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 2

0	1	2	3	4	5	6	7
0x24	0x1f	0x20	0x1f	0x22	0x21	0x1f	0x21
0xf0	0x1f	0xf0	0x1f	0x22	0x21	0x1f	0x21

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

No break here!!

i = 3

0	1	2	3	4	5	6	7
0x24	0x1f	0xf0	0x1f	0x22	0x21	0x1f	0x21
0xf0	0x1f	0xf0	0xff	0x22	0x21	0x1f	0x21

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 4

0	1	2	3	4	5	6	7
0x24	0x1f	0xf0	0xff	0x22	0x21	0x1f	0x21
0xf0	0x1f	0xf0	0xff	0x22	0x0	0x0	0x0

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 5

0	1	2	3	4	5	6	7
0x24	0x1f	0xf0	0xff	0x22	0x0	0x0	0x0
0xf0	0x1f	0xf0	0xff	0x22	0x1f	0x20	0x0

```
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 6

0	1	2	3	4	5	6	7
0x24	0x1f	0xf0	0xff	0x22	0x1f	0x20	0x0
0xf0	0x1f	0xf0	0xff	0x22	0x1f	0xf0	0x0

```c
#define BUF_SIZE 8

void transfer(char *buf) {
  int i;
  for (i = 0; i < BUF_SIZE; i++) {
  switch (buf[i]) {
    case 0x1f:
    if (i > 0) {
      buf[i] += buf[i-1];
    }
    case 0x20: case 0x21:
      buf[i] |= 0xf0;
      break;
    case 0x22: case 0x25:
      *((unsigned *)buf + i/4) = (unsigned)buf[i];
      break;
    case 0x24:
      buf[i] = (char)((*(long *)buf) >> 8);
      break;
    default:
      if (i < BUF_SIZE - 1) {
        *(short *)buf[i] = 0x201f;
      }
      break;
    }
  }
}
```

i = 7

0	1	2	3	4	5	6	7
0x24	0x1f	0xf0	0xff	0x22	0x1f	0xf0	0x0
0xf0	0x1f	0xf0	0xff	0x22	0x1f	0xf0	0x0