

Note:

R3 has the address for the first element

R0 has the address for the last element

1.

```
for(b=90; b<>0; b--) {  
    a[b]=a[b] + z;  
}
```

High-level: First unrolling

```
for(b=90; b<>0; b-=2){  
    a[b] = a[b] + z;  
    a[b-1] = a[b-1] + z;  
}
```

Low-level: First unrolling

```
Loop: LW R1, 0(R0)  
      ADD R1, R1, R2  
      SW R1, 0(R0)  
      LW R4, -8(R0)  
      ADD R4, R4, R2  
      SW R4, -8(R0)  
      ADDI R0, R0, -16  
      BNE R0, R3, Loop
```

High level: Second unrolling

```
for(b=90; b<>0; b-=3){  
    a[b] = a[b] + z;  
    a[b-1] = a[b-1] + z;  
    a[b-2] = a[b-2] + z;  
}
```

Low-level: Second unrolling

```
Loop: LW R1, 0(R0)  
      ADD R1, R1, R2  
      SW R1, 0(R0)  
      LW R4, -8(R0)  
      ADD R4, R4, R2  
      SW R4, -8(R0)  
      LW R5, -16(R0)  
      ADD R5, R5, R2  
      SW R5, -16(R0)
```

```
ADDI R0, R0, -24
BNE R0, R3, Loop
```

2.

Note: R10 has the address for the last element in c array

```
for(z=110; z<>0; z--){
    a[z]=a[z] + c[z];
}
```

High level: First unrolling

```
for(z=110; z<>0; z-=2){
    a[z]= a[z] + c[z];
    a[z-1] = a[z-1] + c[z-1];
}
```

Low-level: First unrolling

```
Loop: LW R1, 0(R0)
      LW R2, 0(R10)
      ADD R1, R1, R2
      SW R1, 0(R0)
      LW R4, -8(R0)
      LW R5, -8(R10)
      ADD R4, R4, R5
      ADDI R0, R0, -16
      BNE R0, R3, Loop
```