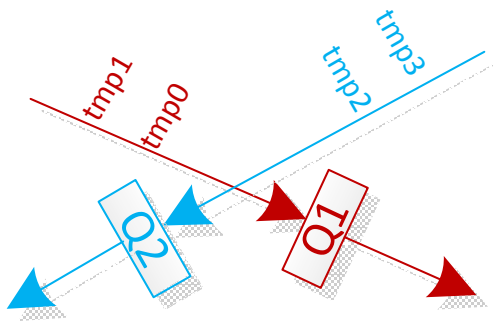
Process 1:

```
int tmp0, tmp1;
int tmp2, tmp 3;
for(int i=0; i<100; i++)
{
  ...
  push(Q1, tmp0);
  push(Q1, tmp1);
  ...
  pop(Q2, tmp2);
  pop(Q2, tmp3);
  ...
}
```



Process 2:

```
int tmp0, tmp1;
int tmp2, tmp 3;
for(int j=0; j<100; j++)
{
  ...
  push(Q2, tmp2);
  push(Q2, tmp3);
  ...
  pop(Q1, tmp0);
  pop(Q1, tmp1);
  ...
}
```

Given Q1 and Q2 both have only ONE buffer slot

- *process 1* cannot push tmp1 until tmp0 is popped by *process 2*
- *process 2* cannot pop tmp0 until tmp3 is pushed
- *process 2* cannot push tmp3 until tmp2 is popped by *process 1*, which happens after tmp1 is pushed – we have a circular dependency

If Q1 and Q2 has infinite size, no circular dependency and no artificial deadlock occurs