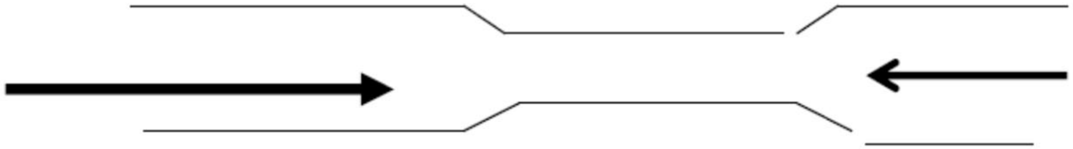
Narrow bridge problem

- For a narrow bridge can only pass cars in one direction . If a car wins the acces to bridge all cars in his direction can pass



- Solution:
- Shared variables:

  Cont_right int = 0;

  Cont_left int = 0;
- Semaphores:

  exmut_right , exmut_left, bridge;
- Initial values for Semaphores = 1

Code Car right

```
P (exmut_right);
  cont_right + +;
  if (cont_right == 1)
  P (bridge);
V (exmut_right);

... He climbs the bridge ...

P (exmut_right);
  contadorsubida --;
  if (cont_right == 0)
  V (bridge);
V (exmut_right);
```

Code Car left

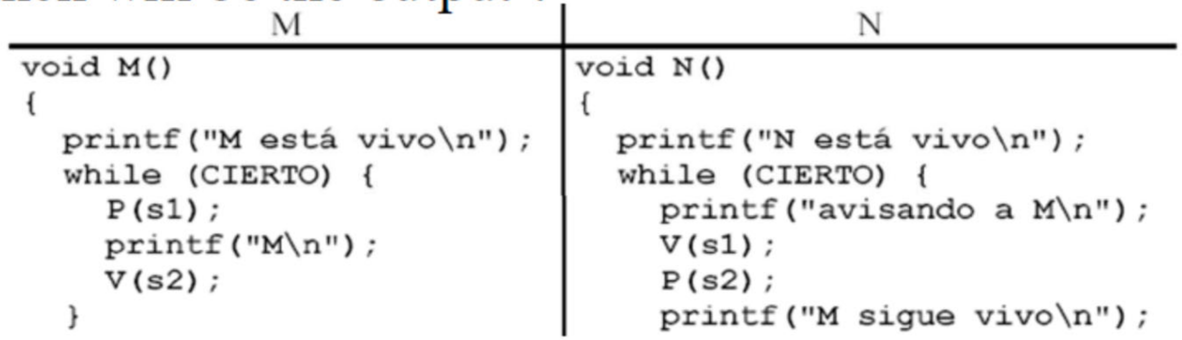
```
P (exmut_left);
  cont_left + +;
  if (cont_left == 1)
  P (bridge);
V (exmut_left);

... Get off the bridge ...

P (exmut_left);
  cont_left --;
  if (contadorbajada == 0)
  V (bridge);
V (exmut_left);
```

S1 and S2 semaphores initialized to 0

Which will be the output ?



M

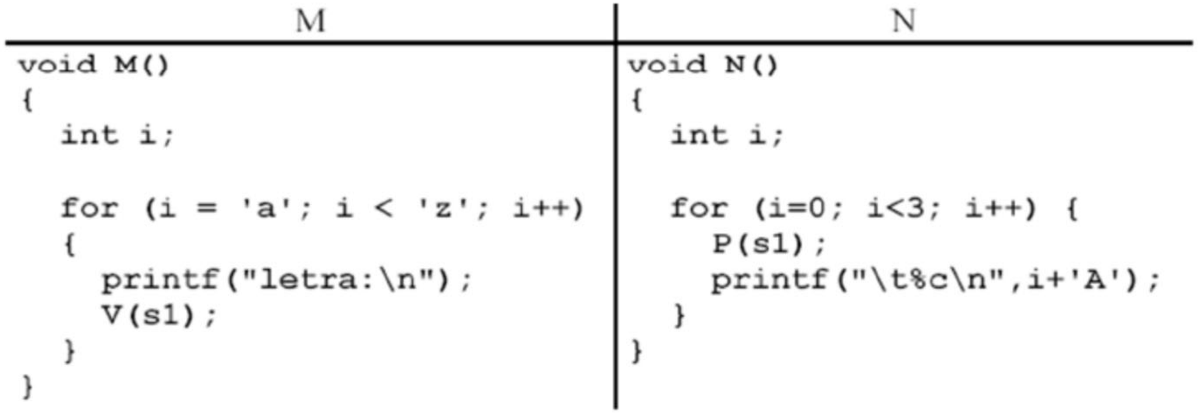
```
void M()
{
  printf("M está vivo\n");
  while (CIERTO) {
    P(s1);
    printf("M\n");
    V(s2);
  }
```

N

```
void N()
{
  printf("N está vivo\n");
  while (CIERTO) {
    printf("avisando a M\n");
    V(s1);
    P(s2);
    printf("M sigue vivo\n");
```

S1 semaphore initialized to 0

Which will see on screen?



M

```
void M()
{
  int i;

  for (i = 'a'; i < 'z'; i++)
  {
    printf("letra:\n");
    V(s1);
  }
}
```

N

```
void N()
{
  int i;

  for (i=0; i<3; i++) {
    P(s1);
    printf("\t%c\n",i+'A');
  }
}
```

Sean A y B dos procesos que se sincronizan mediante los semáforos S_1 y S_2, con valores iniciales 1 y 0, respectivamente. Sabiendo que ejecutan los algorimos que se muestran a continuación, ¿cuáles son las posibles salidas que podrían producirse por la ejecución concurrente de A y B?

A

```
while (CIERTO) {
  int i;

  P(S1);
  for (i=0; i<3; i++) {
    printf("%d", i);
  }
  V(S2);
}
```

B

```
while (CIERTO) {
  int j;

  for (j=3; j<6; j++) {
    P(S2);
    printf("%d", j);
    V(S1);
  }
}
```

Program A	Program B
section A1	section B1
section A2	section B2
Section A3	Section B3

modify the program A and B using semaphores:
A2 not run until you have completed A1 and B1 and
B2 not run until A1 and B1 are finished,
A3 not run until you have completed A2 and B2
and B3 not run until you have completed A2 and B2