```
   adjMat.c


./adjMat 4 1 mat
   mat
4
0 1 0 1
1 0 1 1
0 1 0 1
1 1 1 0
   fir_NC
sample_t fir_NC(sample_t* muestra);


for(i = 0; i<TAP_LENGTH; i++)
   out+=coefsLocalesNC[i] * *(muestra-i);
   TAP_LENGTHTAP_LENGTHTAP_LENGTH









   floatdoublefloatdoubledouble
```

```
for(i = TAP_LENGTH-1; i>0; i--)
   buffer[i]=buffer[i-1];
buffer[0]=muestra;


for(i = 0; i<TAP_LENGTH; i++)
   out+=coefsLocales[i] * buffer[i];
   TAP_LENGTHTAP_LENGTH
   TAP_LENGTH









void update_p(int paso);






   update_pTAP_LENGTHfir(sample_t muestra)TAP_LENGTHTAP_LENGTHupdate_p(int paso)


   update_p




-O

update_p
```

```
void update_p(int paso){
  if(p!=0){
    p = (p+paso) % p;
  }else{
    p+=paso;
  }
}
```

update_ppaso

```
void update_p(){
    if (p+1<0){
        p = TAP_LENGTH + p + 1;
    } else if (p+1>TAP_LENGTH-1){
        p = p + 1 - TAP_LENGTH;
    } else {
        p = p + 1;
    }
}
```

inlinefir.hupdate_p

```
static inline void update_p(int paso);
```

update_pfir

```
gcc -std=c99 -finline-small-functions testfir.o fir.o -o fir_exe
```

-finline-small-functions

```
#define update_p(p){\
    if (p+1<0){\
        p = TAP_LENGTH + p + 1;\
    } else if (p+1>TAP_LENGTH-1){\
        p = p + 1 - TAP_LENGTH;\
    } else {\
        p = p + 1;\
    }\
}
```

```
fir
update_p
```

```
#define update_p(p){ \
    if(p!=0){        \
      p = (p+1) % p; \
    }else{           \
      p += 1;        \
    }                \
}
```