Only difference between child and parent is that child returns a 0 and parent returns 1

Interprocess communication
- Processes within a system may be independent or cooperating
- Cooperating process can affect or be affected by other processes, including sharing data
- Reasons for cooperating processes
  - Information sharing
  - Computation speedup
  - Modularity
  - Convenience
- Cooperating processes need interprocess communication (IPC)
- Two models of IPC
  - Shared memory
  - Message passing

Communications models
- A message passing
  - Both process go through the kernel in a message queue
- Shared message
  - Some portions of memory is mapped to both process, kernel is used to do memory management tricks

Producer consumer problem
- Paradigm for cooperating processes, producer process produces information that is consumed by a consumer process
  - Unbounded-buffer
    - Places no practical limit on the size of the buffer
  - Bounded-buffer
    - Assumes that there is a fixed buffer size
  - Producer process
    - Fills the buffer
  - Consumer process
    - empties the buffer

Bounded buffer shared memory solution
- Shared data

```
#define BUFFER_SIZE 10
typedef struct {
...
} item;

item buffer [BUFFER_SIZE];
int in=0;
int out=0;
```

Bounded buffer producer

```
item next_produced;
while(true){
    /*produce an item in next produced*/
    while(((in+1)%BUFFER_SIZE)==out);
        //do nothing
    buffer[in]=next_produced;
    in=(in+1)%BUFFER_SIZE;
}
```

Bounded buffer consumer

```
item next_consumed;
while(true){
    /*produce an item in next produced*/
    while(in==out);
        //do nothing
    Next_consumed=buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    /* consume the item in next consumed */
}
```

Interprocess communication - message passing

- Mechanism for processes to communicate and to synchronize their actions
- Message system
  - Proocesses communicate with each other ithout resorting

Ring buffer

- At the end of array goes back to begging of array