```
// Program for Slave mode
#include<avr/io.h>
#include<util/delay.h>

void TWI_init_slave(void);
void TWI_match_read_slave(void);
void TWI_read_slave(void);
void TWI_match_write_slave(void);
void TWI_write_slave(void);

unsigned char write_data,recv_data;

int main(void)
{
DDRB=0xff;
TWI_init_slave(); // Function to initilaize slave
while(1)
{
TWI_match_read_slave(); //Function to match the slave address and slave dirction bit(read)
TWI_read_slave(); // Function to read data

write_data=~recv_data; // Togglem the receive data

TWI_match_write_slave(); //Function to match the slave address and slave dirction bit(write)
TWI_write_slave();       // Function to write data
}
}

void TWI_init_slave(void) // Function to initilaize slave
{
TWAR=0x20; // Fill slave address to TWAR
}

void TWI_write_slave(void) // Function to write data
{
TWDR= write_data;           // Fill TWDR register whith the data to be sent
TWCR= (1<<TWEN)|(1<<TWINT);   // Enable TWI, Clear TWI interrupt flag
while((TWSR & 0xF8) != 0xC0); // Wait for the acknowledgement
}

void TWI_match_write_slave(void) //Function to match the slave address and slave dirction
bit(write)
{
while((TWSR & 0xF8)!= 0xA8) // Loop till correct acknoledgement have been received
{
// Get acknowlegement, Enable TWI, Clear TWI interrupt flag
TWCR=(1<<TWEA)|(1<<TWEN)|(1<<TWINT);
while (!(TWCR & (1<<TWINT)));  // Wait for TWINT flag
}
}

void TWI_read_slave(void)
{
// Clear TWI interrupt flag,Get acknowlegement, Enable TWI
TWCR= (1<<TWINT)|(1<<TWEA)|(1<<TWEN);
while (!(TWCR & (1<<TWINT))); // Wait for TWINT flag
```

```
while((TWSR & 0xF8)!=0x80); // Wait for acknowledgement
recv_data=TWDR; // Get value from TWDR
PORTB=recv_data; // send the receive value on PORTB
}

void TWI_match_read_slave(void) //Function to match the slave address and slave dirction
bit(read)
{
while((TWSR & 0xF8)!= 0x60)  // Loop till correct acknoledgement have been received
{
// Get acknowlegement, Enable TWI, Clear TWI interrupt flag
TWCR=(1<<TWEA)|(1<<TWEN)|(1<<TWINT);
while (!(TWCR & (1<<TWINT)));  // Wait for TWINT flag
}
}
```