

```

/*****
*****
*****

```

Start this is where you set up  
the message you want display  
7-10 characters

```

*****
*****
*****

```

```
// setup
```

```
String messageF = "MUZZLED";
```

```
//PUT YOUR MESSAGE HERE!!-
```

must be in all caps, spaces  
are fine, no punctuation

```
String messageR = "DELZZUM";
```

```
//PUT the reverse of your  
message here!
```

```
byte refreshrate = 2;//delay
```

time for pixels to refresh in  
milliseconds- experiment with  
different values

```
int textLen1 = messageF.  
length(); // gets the length  
of messageF  
int textLen2 = messageR.  
length(); // gets the length  
of messageR
```

```
//variable initialization  
byte storage1 = 0; //for portB  
byte storage2 = 0; //for portC  
byte storage3 = 0; //for portD  
byte the; //  
byte n; //variable for loops  
byte t; //variable for loops
```

```
byte 1; //variable for loops
```

```
const boolean A[] PROGMEM = {
```

```
// Letter A
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,1,1,1,1,1,1,0,0,0,
```

```
  0,0,1,1,1,1,1,1,1,1,0,0,
```

```
  0,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,0,0,1,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
};
```

```
const boolean B[]  PROGMEM = {  
//Letter B
```

```
1,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,
```

```
1,1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,0,0,  
};
```

```
const boolean C[]  PROGMEM = {  
//Letter C  
0,0,0,0,1,1,1,1,1,1,1,0,  
0,0,1,1,1,1,1,1,1,1,1,1,
```

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,

```
} ;
```

```
const boolean D[]  PROGMEM = {
```

```
//Letter D
```

```
  1,1,1,1,1,1,1,1,1,0,0,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,0,0,0,1,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,0,0,0,  
};
```

```
const boolean E[]  PROGMEM = {  
//Letter E
```

```
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,0,0,0,0,0,0,0,0,  
1,1,1,1,0,0,0,0,0,0,0,0,  
1,1,1,1,0,0,0,0,0,0,0,0,  
1,1,1,1,0,0,0,0,0,0,0,0,
```



```
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
};
```

```
const boolean F[]  PROGMEM = {
//Letter F
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
```

[illegible]

```
} ;
```

```
const boolean G[]  PROGMEM = {
```

```
//Letter G
```

```
  0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,
```

```
  0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
```

```
  0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,
```

[illegible]

```
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,1,
```

```
};
```

```
const boolean I[]  PROGMEM = {
//Letter I
```

```
1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

```
};
```

```
const boolean J[]  PROGMEM = {
```

```
//Letter J
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
0,0,0,0,1,1,1,1,0,0,0,0,  
1,1,1,0,1,1,1,1,0,0,0,0,  
1,1,1,1,1,1,1,1,0,0,0,0,  
1,1,1,1,1,1,1,1,0,0,0,0,  
0,1,1,1,1,1,1,0,0,0,0,0,  
0,0,1,1,1,1,0,0,0,0,0,0,  
};
```

```
const boolean K[]  PROGMEM = {  
//Letter K
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,1,1,1,1,1,0,  
1,1,1,1,0,1,1,1,1,1,1,0,
```



```
1,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,1,1,1,1,1,0,0,0,  
1,1,1,1,1,1,1,1,0,0,0,0,  
1,1,1,1,1,1,1,1,1,0,0,0,  
1,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,0,1,1,1,1,1,0,0,  
1,1,1,1,0,0,1,1,1,1,1,0,  
1,1,1,1,0,0,0,1,1,1,1,0,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
};
```

```
const boolean L[]  PROGMEM = {  
//Letter L  
1,1,1,1,0,0,0,0,0,0,0,0,  
1,1,1,1,0,0,0,0,0,0,0,0,
```

[illegible]

```
} ;
```

```
const boolean M[]  PROGMEM = {
```

```
//Letter M
```

```
  1,1,1,0,0,0,0,0,0,1,1,1,
```

```
  1,1,1,0,0,0,0,0,0,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,1,0,0,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,0,1,1,1,1,0,1,1,1,
```

```
  1,1,1,0,1,1,1,1,0,1,1,1,
```

```
  1,1,1,0,0,1,1,0,0,1,1,1,
```

```
  1,1,1,0,0,1,1,0,0,1,1,1,
```

```
  1,1,1,0,0,0,0,0,0,1,1,1,
```

```
  1,1,1,0,0,0,0,0,0,1,1,1,
```

```
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
};
```

```
const boolean N[]  PROGMEM = {  
//Letter N
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,1,0,0,0,1,1,1,1,  
1,1,1,1,1,0,0,0,1,1,1,1,  
1,1,1,1,1,0,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,
```

```
1,1,1,1,1,1,1,0,1,1,1,1,  
1,1,1,1,1,1,1,0,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,0,1,1,1,1,1,1,1,  
1,1,1,1,0,1,1,1,1,1,1,1,  
1,1,1,1,0,0,1,1,1,1,1,1,  
1,1,1,1,0,0,1,1,1,1,1,1,  
1,1,1,1,0,0,1,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,
```

```
};
```

```
const boolean O[]  PROGMEM = {  
//Letter O
```

```
0,0,0,1,1,1,1,1,1,0,0,0,  
0,1,1,1,1,1,1,1,1,1,1,0,
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

```
} ;
```

```
const boolean P[]  PROGMEM = {
```

```
//Letter P
```

```
  1,1,1,1,1,1,1,1,1,1,0,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,1,
```

```
  1,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,1,1,1,1,1,1,0,
```

```
  1,1,1,1,1,1,1,1,1,1,0,0,
```

```
  1,1,1,1,0,0,0,0,0,0,0,0,
```

```
  1,1,1,1,0,0,0,0,0,0,0,0,
```

```
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,
};
```

```
const boolean Q[]  PROGMEM = {
//Letter Q
0,0,0,1,1,1,1,1,1,0,0,0,
0,1,1,1,1,1,1,1,1,1,1,0,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,
```



```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,1,0,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,0,  
0,1,1,1,1,1,1,1,1,1,1,1,  
0,0,0,1,1,1,1,1,0,1,1,1,  
};
```

```
const boolean R[]  PROGMEM = {  
//Letter R  
1,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,1,1,1,1,1,1,1,0,
```



```
} ;
```

```
const boolean S[]  PROGMEM = {
```

```
//Letter S
```

```
  0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
```

```
  0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
```

```
  0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
```

```
  0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,
```

```
  0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,
```

```
  0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0,
```

```
  0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
```

```
  0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
```

```
0,0,0,0,0,0,0,1,1,1,1,1,
0,0,0,0,0,0,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,0,
};
```

```
const boolean T[]  PROGMEM = {
//Letter T
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
0,0,0,0,1,1,1,1,0,0,0,0,
0,0,0,0,1,1,1,1,0,0,0,0,
0,0,0,0,1,1,1,1,0,0,0,0,
0,0,0,0,1,1,1,1,0,0,0,0,
```

```
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,
```

```
};
```

```
const boolean U[]  PROGMEM = {  
//Letter U
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,
```

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

```
} ;
```

```
const boolean V[]  PROGMEM = {
```

```
//Letter V
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  0,1,1,1,0,0,0,0,1,1,1,0,
```

```
  0,1,1,1,1,0,0,1,1,1,1,0,
```

```

0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,
} ;

```

```
const boolean W[]  PROGMEM = {
//Letter W
```

[illegible]



```
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,0,0,0,0,1,1,1,  
1,1,1,0,0,1,1,0,0,1,1,1,  
1,1,1,0,0,1,1,0,0,1,1,1,  
1,1,1,0,0,1,1,0,0,1,1,1,  
1,1,1,0,1,1,1,1,0,1,1,1,  
1,1,1,0,1,1,1,1,0,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,0,  
0,1,1,1,1,1,1,1,1,1,1,0,  
0,1,1,1,1,0,0,1,1,1,1,0,  
};
```

```
const boolean X[]  PROGMEM = {  
//Letter X  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,
```

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,  
0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,  
0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,  
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,  
0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,  
0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

```
} ;
```

```
const boolean Y[]  PROGMEM = {
```

```
//Letter Y
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  1,1,1,1,0,0,0,0,1,1,1,1,
```

```
  0,1,1,1,0,0,0,0,1,1,1,0,
```

```
  0,1,1,1,1,0,0,1,1,1,1,0,
```

```
  0,0,1,1,1,0,0,1,1,1,0,0,
```

```
  0,0,1,1,1,1,1,1,1,1,0,0,
```

```
  0,0,0,1,1,1,1,1,1,0,0,0,
```

```
  0,0,0,1,1,1,1,1,1,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
  0,0,0,0,1,1,1,1,0,0,0,0,
```

```
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,0,0,0,0,  
};
```

```
const boolean Z[] PROGMEM = {  
//Letter Z  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
0,0,0,0,0,0,0,1,1,1,1,0,  
0,0,0,0,0,0,1,1,1,1,1,0,  
0,0,0,0,0,0,1,1,1,1,0,0,  
0,0,0,0,0,1,1,1,1,1,0,0,
```

```
0,0,0,0,0,1,1,1,1,0,0,0,
0,0,0,0,0,1,1,1,1,0,0,0,
0,0,0,0,1,1,1,1,0,0,0,0,
0,0,0,1,1,1,1,1,0,0,0,0,
0,0,0,1,1,1,1,0,0,0,0,0,
0,0,1,1,1,1,1,0,0,0,0,0,
0,1,1,1,1,1,0,0,0,0,0,0,
0,1,1,1,1,0,0,0,0,0,0,0,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
};
```

```
const boolean B2[] PROGMEM {
//Letter Reverse B
0,0,1,1,1,1,1,1,1,1,1,1,
0,1,1,1,1,1,1,1,1,1,1,1,
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

} ;

```
const boolean C2[]  PROGMEM {  
//Letter Reverse C  
  0,1,1,1,1,1,1,1,0,0,0,0,  
  1,1,1,1,1,1,1,1,1,1,0,0,  
  1,1,1,1,1,1,1,1,1,1,1,0,  
  1,1,1,1,1,1,1,1,1,1,1,0,  
  1,1,0,0,0,0,1,1,1,1,1,1,  
  0,0,0,0,0,0,0,1,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,1,1,1,1,1,  
  1,1,0,0,0,0,1,1,1,1,1,1,
```

```
1,1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,0,0,  
0,1,1,1,1,1,1,1,0,0,0,0,  
};
```

```
const boolean D2[] PROGMEM {  
//Letter Reverse D
```

```
0,0,0,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,0,1,1,1,1,1,  
1,1,1,1,1,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,
```



```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,1,0,1,0,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,  
0,0,0,1,1,1,1,1,1,1,1,1,  
};
```

```
const boolean E2[]  PROGMEM {  
//Letter Reverse E
```

```
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,
```

0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

} ;

```
const boolean F2[]  PROGMEM {  
//Letter Reverse F  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
}
```

```
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
};
```

```
const boolean G2[] PROGMEM {  
//Letter Reverse G
```

```
0,0,0,1,1,1,1,1,0,0,0,0,  
0,1,1,1,1,1,1,1,1,1,0,0,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,0,0,1,1,1,1,1,1,  
0,0,0,0,0,0,0,1,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,
```

```
1,1,1,1,1,1,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,  
1,1,1,1,1,1,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,1,1,1,1,1,  
1,1,1,1,0,0,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,0,  
0,1,1,1,1,1,1,1,1,1,0,0,  
0,0,0,1,1,1,1,1,0,0,0,0,  
};
```

```
const boolean J2[] PROGMEM {  
//Letter Reverse J
```

```
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,
```

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,  
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0,  
0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0,

} ;

```
const boolean K2[]  PROGMEM {  
//Letter Reverse K  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,1,0,0,0,1,1,1,1,  
  1,1,1,1,1,0,0,0,1,1,1,1,  
  0,1,1,1,1,1,0,0,1,1,1,1,  
  0,1,1,1,1,1,1,0,1,1,1,1,  
  0,0,1,1,1,1,1,1,1,1,1,1,  
  0,0,0,1,1,1,1,1,1,1,1,1,  
  0,0,0,0,1,1,1,1,1,1,1,1,  
  0,0,0,1,1,1,1,1,1,1,1,1,  
  0,0,1,1,1,1,1,1,1,1,1,1,  
  0,0,1,1,1,1,1,0,1,1,1,1,  
  0,1,1,1,1,1,0,0,1,1,1,1,  
  0,1,1,1,1,0,0,0,1,1,1,1,
```

} ;

//Letter Reverse L

0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,



```
0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,
};
```

```
const boolean N2[]  PROGMEM {
//Letter Reverse N
1,1,1,1,0,0,0,0,1,1,1,1,
1,1,1,1,0,0,0,0,1,1,1,1,
1,1,1,1,0,0,0,1,1,1,1,1,
1,1,1,1,0,0,0,1,1,1,1,1,
```

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

} ;

```
const boolean P2[]  PROGMEM {  
//Letter Reverse P  
  0,0,1,1,1,1,1,1,1,1,1,1,  
  0,1,1,1,1,1,1,1,1,1,1,1,  
  0,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,0,0,0,0,1,1,1,1,  
  1,1,1,1,1,1,1,1,1,1,1,1,  
  0,1,1,1,1,1,1,1,1,1,1,1,  
  0,1,1,1,1,1,1,1,1,1,1,1,  
  0,0,1,1,1,1,1,1,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
  0,0,0,0,0,0,0,0,1,1,1,1,  
}
```

```
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
0,0,0,0,0,0,0,0,1,1,1,1,  
};
```

```
const boolean Q2[] PROGMEM {  
//Letter Reverse Q
```

```
0,0,0,1,1,1,1,1,1,0,0,0,  
0,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,0,0,1,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,
```

```
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,0,0,0,0,1,1,1,1,  
1,1,1,1,1,0,0,0,1,1,1,1,  
1,1,1,1,1,1,0,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,0,  
1,1,1,0,1,1,1,1,1,0,0,0,  
};
```

```
const boolean R2[] PROGMEM {  
// Letter Reverse R  
0,0,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,
```

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,  
1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

} ;

```
const boolean S2[] PROGMEM {  
  // Letter Reverse S  
    1,1,1,1,1,1,1,1,1,1,0,0,  
    1,1,1,1,1,1,1,1,1,1,1,0,  
    1,1,1,1,1,1,1,1,1,1,1,0,  
    1,1,1,1,1,1,1,1,1,1,1,1,  
    0,0,0,0,0,0,0,0,1,1,1,1,  
    0,0,0,0,0,0,0,0,1,1,1,1,  
    0,0,0,0,0,0,0,1,1,1,1,1,  
    0,0,0,0,0,0,1,1,1,1,1,0,  
    0,0,0,0,0,1,1,1,1,1,0,0,  
    0,0,0,0,1,1,1,1,1,0,0,0,  
    0,0,0,1,1,1,1,1,0,0,0,0,  
    0,0,1,1,1,1,1,0,0,0,0,0,  
    0,1,1,1,1,1,0,0,0,0,0,0,  
    1,1,1,1,1,1,0,0,0,0,0,0,  
    1,1,1,1,1,0,0,0,0,0,0,0,  
    1,1,1,1,1,1,0,0,0,0,0,0,
```

```
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,1,1,1,1,1,1,1,1,  
};
```

```
const boolean Z2[] PROGMEM {  
// Letter Reverse Z
```

```
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,1,  
0,1,1,1,1,0,0,0,0,0,0,0,0,  
0,1,1,1,1,1,0,0,0,0,0,0,0,  
0,0,1,1,1,1,0,0,0,0,0,0,0,  
0,0,1,1,1,1,1,0,0,0,0,0,0,  
0,0,0,1,1,1,1,0,0,0,0,0,0,  
0,0,0,1,1,1,1,0,0,0,0,0,0,
```



```
0,0,0,0,1,1,1,1,0,0,0,0,  
0,0,0,0,1,1,1,1,1,0,0,0,  
0,0,0,0,0,1,1,1,1,0,0,0,  
0,0,0,0,0,1,1,1,1,1,0,0,  
0,0,0,0,0,0,1,1,1,1,1,0,  
0,0,0,0,0,0,0,1,1,1,1,0,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,1,1,
```

```
};
```

```
void print1(boolean  
letterArray[]) { //function  
print out the array letters
```

```
    for (t=0; t<12; t++){ //for
each of the columns in the
letter arrays

        for (l=0; l<6; l++){ //for
first six rows of in the
letter storage

            storage1 = storage1 << 1;
//bitwise shift left
            storage1 |=
pgm_read_byte_near(letterArray
+ (l*12+t)); //add next value
from storageset
        }

        for (l=6; l<14; l++){
//for next eight rows of
letter storage

            storage2 = storage2 << 1;
```

```
//bitwise shift left
    storage2 |=
pgm_read_byte_near(letterArray
+ (l*12+t)); //add next value
from storageset
    }
    for (l=14; l<20; l++){
//for next six rows of letter
storage
        storage3 = storage3 << 1;
//bitwise shift left
        storage3 |=
pgm_read_byte_near(letterArray
+ (l*12+t)); //add next value
from storageset
    }
    //SET PINS:
    PORTB = storage1;
```

```
PORTD = storage2;  
PORTC = storage3;  
delay(refreshrate);
```

```
//clear storage  
storage1=0;  
storage2=0;  
storage3=0;  
}
```

```
}
```

```
void setup() {  
    // Serial.begin(9600); //  
used for testing and debugging  
    DDRB = 0xFF; //port b-  
digital pins 8-13  
    DDRD = 0xFF; //port d-  
digital pins 0-7
```

```
// assigning pinmodes to  
analog pins
```

```
pinMode(A0, OUTPUT);
```

```
pinMode(A1, OUTPUT);
```

```
pinMode(A2, OUTPUT);
```

```
pinMode(A3, OUTPUT);
```

```
pinMode(A5, INPUT);
```

```
delay(500);
```

```
}
```

```
void loop() {
```

```
// reads the potentiometer  
values and the adjusts the
```

```
// blinking speed of LEDs so  
users can change the speed
```

```
int sensorValue =  
analogRead(A5);  
float refreshrate =  
sensorValue * (5.0 / 256); //  
goes from 0-20 for speed  
//Serial.  
println(refreshrate); // used  
for testing and debugging  
  
//space at beginning of text  
PORTB = 0;  
PORTD = 0;  
PORTC = 0;  
delay(refreshrate*3);  
  
// this for loop with be  
responsible for looping  
through the word that the user
```

specifies

```
    for (n=0; n<textLen1; n++)  
{//go through each character  
of the user inputted text and  
call function print1 to  
display letter  
    if (messageF.  
charAt(n)=='A') {  
        print1(A);  
    }  
    else if (messageF.  
charAt(n)=='B') {  
        print1(B);  
    }  
    else if (messageF.  
charAt(n)=='C') {  
        print1(C);  
    }
```

```
        else if (messageF.  
charAt (n) == 'D' ) {  
            print1 (D) ;  
        }  
        else if (messageF.  
charAt (n) == 'E' ) {  
            print1 (E) ;  
        }  
        else if (messageF.  
charAt (n) == 'F' ) {  
            print1 (F) ;  
        }  
        else if (messageF.  
charAt (n) == 'G' ) {  
            print1 (G) ;  
        }  
        else if (messageF.  
charAt (n) == 'H' ) {
```



```
        print1 (H) ;
    }
    else if (messageF.
charAt (n) == 'I' ) {
        print1 (I) ;
    }
    else if (messageF.
charAt (n) == 'J' ) {
        print1 (J) ;
    }
    else if (messageF.
charAt (n) == 'K' ) {
        print1 (K) ;
    }
    else if (messageF.
charAt (n) == 'L' ) {
        print1 (L) ;
    }
```

```
        else if (messageF.  
charAt (n) == 'M' ) {  
            print1 (M) ;  
        }  
        else if (messageF.  
charAt (n) == 'N' ) {  
            print1 (N) ;  
        }  
        else if (messageF.  
charAt (n) == 'O' ) {  
            print1 (O) ;  
        }  
        else if (messageF.  
charAt (n) == 'P' ) {  
            print1 (P) ;  
        }  
        else if (messageF.  
charAt (n) == 'Q' ) {
```

```
        print1 (Q) ;
    }
    else if (messageF.
charAt (n) == 'R' ) {
        print1 (R) ;
    }
    else if (messageF.
charAt (n) == 'S' ) {
        print1 (S) ;
    }
    else if (messageF.
charAt (n) == 'T' ) {
        print1 (T) ;
    }
    else if (messageF.
charAt (n) == 'U' ) {
        print1 (U) ;
    }
```

```
        else if (messageF.  
charAt (n) == 'V' ) {  
            print1 (V) ;  
        }  
        else if (messageF.  
charAt (n) == 'W' ) {  
            print1 (W) ;  
        }  
        else if (messageF.  
charAt (n) == 'X' ) {  
            print1 (X) ;  
        }  
        else if (messageF.  
charAt (n) == 'Y' ) {  
            print1 (Y) ;  
        }  
        else if (messageF.  
charAt (n) == 'Z' ) {
```

```
        print1(Z);
    }
    else if (messageF.
charAt(n)==' ') {
        PORTB = 0;
        PORTD = 0;
        PORTC = 0;
        delay(refreshrate*3); //off
for 3 pixels
    }
    //space between each
character
        PORTB = 0;
        PORTD = 0;
        PORTC = 0;
        delay(refreshrate);
    }
    //space at end of text
```

```
PORTB = 0;
PORTD = 0;
PORTC = 0;
delay(refreshrate*3);
```

```
    delay(20); // this delay
will time it so that when
returning the letter will
print correctly
```

```

/*****
*****
*** /
```

```
    // Reverse printing //
```

```
    // this for loop will run
through the reverse word that
```

the user specifies in order to  
get the correct

// word orientation when  
the user flips the wand  
backwards

```
for (n=0; n<textLen2; n++)  
{//go through each character  
of Reverse message and call  
function print1 to display  
letter
```

```
    if (messageR.  
charAt (n) == 'A' ) {  
        print1 (A) ;  
    }
```

```
    else if (messageR.  
charAt (n) == 'B' ) {  
        print1 (B2) ;  
    }
```

```
        else if (messageR.  
charAt (n) == 'C') {  
            print1 (C2) ;  
        }  
        else if (messageR.  
charAt (n) == 'D') {  
            print1 (D2) ;  
        }  
        else if (messageR.  
charAt (n) == 'E') {  
            print1 (E2) ;  
        }  
        else if (messageR.  
charAt (n) == 'F') {  
            print1 (F2) ;  
        }  
        else if (messageR.  
charAt (n) == 'G') {
```



```
        print1 (G2) ;
    }
    else if (messageR.
charAt (n) == 'H' ) {
        print1 (H) ;
    }
    else if (messageR.
charAt (n) == 'I' ) {
        print1 (I) ;
    }
    else if (messageR.
charAt (n) == 'J' ) {
        print1 (J2) ;
    }
    else if (messageR.
charAt (n) == 'K' ) {
        print1 (K2) ;
    }
```

```
        else if (messageR.  
charAt (n) == 'L' ) {  
            print1 (L2) ;  
        }  
        else if (messageR.  
charAt (n) == 'M' ) {  
            print1 (M) ;  
        }  
        else if (messageR.  
charAt (n) == 'N' ) {  
            print1 (N2) ;  
        }  
        else if (messageR.  
charAt (n) == 'O' ) {  
            print1 (O) ;  
        }  
        else if (messageR.  
charAt (n) == 'P' ) {
```

```
        print1 (P2) ;
    }
    else if (messageR.
charAt (n) == 'Q' ) {
        print1 (Q2) ;
    }
    else if (messageR.
charAt (n) == 'R' ) {
        print1 (R2) ;
    }
    else if (messageR.
charAt (n) == 'S' ) {
        print1 (S2) ;
    }
    else if (messageR.
charAt (n) == 'T' ) {
        print1 (T) ;
    }
```

```
        else if (messageR.  
charAt (n) == 'U' ) {  
            print1 (U) ;  
        }  
        else if (messageR.  
charAt (n) == 'V' ) {  
            print1 (V) ;  
        }  
        else if (messageR.  
charAt (n) == 'W' ) {  
            print1 (W) ;  
        }  
        else if (messageR.  
charAt (n) == 'X' ) {  
            print1 (X) ;  
        }  
        else if (messageR.  
charAt (n) == 'Y' ) {
```

```

        print1(Y);
    }
    else if (messageR.
charAt(n)=='Z') {
        print1(Z2);
    }
    else if (messageR.
charAt(n)==' ') {
        PORTB = 0;
        PORTD = 0;
        PORTC = 0;

        delay(refreshrate*3); //off
for 3 pixels
    }

    //space between each
character

    PORTB = 0;
    PORTD = 0;

```

```
    PORTC = 0;
    delay(refreshrate);
}
//space at end of text
    PORTB = 0;
    PORTD = 0;
    PORTC = 0;
    delay(refreshrate*3);

    delay(20); // this delay
will be responsible for timing
the foward letter correctly
}
```