```
//write LPC1768 c to blionk LED CONNECTED TO  p1.19PIN
//FOREVER
#include <LPC17xx.h>
#define LED1 (0x01 << 19)

int main()
{
int i;
//CONFIGURE p1.19 PIN AS OUTPUT PIN
LPC_GPIO1->FIODIR |= LED1 ;
   //
while(1){
   LPC_GPIO1->FIOSET=LED1; //turn on LED
   for(i=0;i<100000;i++){}
   LPC_GPIO1->FIOCLR=LED1; //turn off LED
   for(i=0;i<100000;i++){}
}
}

#include <LPC17xx.h>

#define LED1 (0x01 << 19)       // Define LED1 connected to P1.19
#define ALL_LED (0xFF << 19)    // Define all LEDs connected from P1.19 to P1.26

void delay_ms(uint32_t millis);

int main(void)
{
   // Configure P1.19 - P1.26 as output pins
   LPC_GPIO1->FIODIR |= ALL_LED; // Set P1.19 to P1.26 as outputs
   LPC_GPIO1->FIOCLR = ALL_LED;  // Turn off all LEDs initially
```

```
    while (1) {
        LPC_GPIO1->FIOSET = ALL_LED;  // Turn on all LEDs
        delay_ms(1000);               // 1-second delay
        LPC_GPIO1->FIOCLR = ALL_LED;  // Turn off all LEDs
        delay_ms(1000);               // 1-second delay
    }
}

void delay_ms(uint32_t millis)
{
    uint32_t i, j;
    for (i = 0; i < millis; i++) {
        for (j = 0; j < 1250; j++) {
            // Small time delay loop
        }
    }
}
```

```
#include <LPC17xx.h>

#define ALL_LED (0xFF << 19)    // Define all LEDs connected from P1.19 to P1.26

void delay_ms(uint32_t millis);

int main(void)
{
    uint32_t i;  // Declare the loop variable at the beginning of the block

    // Configure P1.19 - P1.26 as output pins
```

```
    LPC_GPIO1->FIODIR |= ALL_LED; // Set P1.19 to P1.26 as outputs
    LPC_GPIO1->FIOCLR = ALL_LED;  // Turn off all LEDs initially

    while (1) {
        for (i = 19; i <= 26; i++) {  // Iterate through LEDs from P1.19 to P1.26
            LPC_GPIO1->FIOSET = (1 << i);      // Turn on the current LED
            delay_ms(500);                  // 500 ms delay
            LPC_GPIO1->FIOCLR = (1 << i);     // Turn off the current LED
        }
    }
}

void delay_ms(uint32_t millis)
{
    uint32_t i, j;
    for (i = 0; i < millis; i++) {
        for (j = 0; j < 1250; j++) {
            // Small time delay loop
        }
    }
}



#include <LPC17xx.h>

#define ALL_LED (0xFF << 19)    // Define all LEDs connected from P1.19 to P1.26

void delay_ms(uint32_t millis);

int main(void)
{
```

```
    uint32_t i;  // Declare the loop variable at the beginning of the block

    // Configure P1.19 - P1.26 as output pins
    LPC_GPIO1->FIODIR |= ALL_LED; // Set P1.19 to P1.26 as outputs
    LPC_GPIO1->FIOCLR = ALL_LED;  // Turn off all LEDs initially

    while (1) {
        // First phase: Sequentially light up LEDs from P1.19 to P1.26
        for (i = 19; i <= 26; i++) {
            LPC_GPIO1->FIOSET = (1 << i);      // Turn on the current LED
            delay_ms(200);                     // 200 ms delay
        }

        // Second phase: Sequentially light up LEDs from P1.26 to P1.19
        for (i = 26; i >= 19; i--) {
            LPC_GPIO1->FIOSET = (1 << i);      // Turn on the current LED
            delay_ms(200);                     // 200 ms delay
        }

        // Brief pause before restarting the cycle
        LPC_GPIO1->FIOCLR = ALL_LED;
        delay_ms(500); // 500 ms delay before starting again
    }
}

void delay_ms(uint32_t millis)
{
    uint32_t i, j;
    for (i = 0; i < millis; i++) {
        for (j = 0; j < 1250; j++) {
            // Small time delay loop
```

```
    }
  }
}

#include <LPC17xx.h>

#define ALL_LED (0xFF << 19) // Define all LEDs connected from P1.19 to P1.26

void delay_ms(uint32_t millis);

int main(void)
{
  uint32_t i, mid = 22; // Center LED index (P1.22 is the midpoint for P1.19 to P1.26)

  // Configure P1.19 - P1.26 as output pins
  LPC_GPIO1->FIODIR |= ALL_LED; // Set P1.19 to P1.26 as outputs
  LPC_GPIO1->FIOCLR = ALL_LED;  // Turn off all LEDs initially

  while (1) {
    // Converging effect: LEDs light up from outer edges to the center
    for (i = 0; i <= (mid - 19); i++) {
      LPC_GPIO1->FIOSET = (1 << (19 + i)) | (1 << (26 - i)); // Turn on symmetrical LEDs
      delay_ms(50); // Delay for convergence
    }

    delay_ms(50); // Hold all LEDs ON at the center

    // Diverging effect: LEDs light up from the center outward (reverse of converging)
    for (i = 0; i <= (mid - 19); i++) {
      LPC_GPIO1->FIOCLR = (1 << (mid - i)) | (1 << (mid + i)); // Turn off LEDs symmetrically inward
      delay_ms(50); // Delay for divergence (moving outward)
```

```
        }

        delay_ms(300); // Hold all LEDs OFF after divergence

        // Reset LEDs to OFF before restarting the cycle
        LPC_GPIO1->FIOCLR = ALL_LED;
        delay_ms(50); // Brief pause before the next cycle
    }
}

void delay_ms(uint32_t millis)
{
    uint32_t i, j;
    for (i = 0; i < millis; i++) {
        for (j = 0; j < 1250; j++) {
            // Small time delay loop
        }
    }
}



#include <LPC17xx.h>
 // LEDs connected from P1.19 to P1.26
#define ALL_LED (0xFF << 19)
#define GROUP1 (0x0F << 19)
#define GROUP2 (0x0F << 23)
void delay_ms(uint32_t millis);

int main(void)
{
    //uint32_t i;
```

```
    // Configure P1.19 - P1.26 as output pins

    LPC_GPIO1->FIODIR |= ALL_LED; // Set P1.19 to P1.26 as outp
    LPC_GPIO1->FIOCLR = ALL_LED;

    while (1) {
          LPC_GPIO1->FIOSET = GROUP1;
          LPC_GPIO1->FIOCLR = GROUP2;
       delay_ms(5);
     LPC_GPIO1->FIOSET = GROUP2;
     LPC_GPIO1->FIOCLR = GROUP1;
     delay_ms(5);
    }
}

void delay_ms(uint32_t millis)
{
    uint32_t i, j;
    for (i = 0; i < millis; i++) {
       for (j = 0; j < 1250; j++) {
          // Small time delay loop
       }
    }
}

# include <LPC17xx.h>
#define BUZZER (0x01 << 27)

void delay_ms(uint32_t mls){
uint32_t i, j;
```

```
for (i = 0; i < mls; i++){
for (j = 0; j < 1250; j++){}
}
}

int main(){
LPC_GPIO1 -> FIODIR |= BUZZER;
LPC_GPIO1 -> FIOCLR = BUZZER;

while(1){
      LPC_GPIO1 -> FIOSET = BUZZER;
      delay_ms(250);

      LPC_GPIO1 -> FIOCLR = BUZZER;
      delay_ms(250);
   }
}

# include <LPC17xx.h>
# include <stdlib.h>
# include <time.h>
#define BUZZER (0x01 << 27)

void delay_ms(uint32_t mls){
uint32_t i, j;
for (i = 0; i < mls; i++){
for (j = 0; j < 1250; j++){}
}
}

int main(){
```

```
uint32_t i;
LPC_GPIO1 -> FIODIR |= BUZZER;
LPC_GPIO1 -> FIOCLR = BUZZER;

for (i = 0; i < 10; i++){
      int rand_val = (rand() % (1000 - 100 + 1));

      LPC_GPIO1 -> FIOSET = BUZZER;
      delay_ms(rand_val);

      LPC_GPIO1 -> FIOCLR = BUZZER;
      delay_ms(rand_val);

   }
}

# include <LPC17xx.h>
# include <stdint.h>
#define BUZZER (1 << 27)
#define SWITCH (1 << 11)

void delay(uint32_t ms){
uint32_t i, j;
   for(i = 0; i < ms; i++){
      for(j = 0; j < 1250; j++){}
}
}

int main(){
   LPC_GPIO1 -> FIODIR |= BUZZER; //OUTPUT KE LIYE OR
   LPC_GPIO2 -> FIODIR &= ~SWITCH;//INPUT KE LIYE AND
```

```
   LPC_GPIO1 -> FIOCLR = BUZZER;

   while(1){
      if ((LPC_GPIO2 -> FIOPIN & SWITCH) != 0){
         LPC_GPIO1 -> FIOSET = BUZZER;
      }

      else{
         LPC_GPIO1 -> FIOCLR = BUZZER;
      }
   }

}

# include <LPC17xx.h>
# include <stdint.h>
#define BUZZER (1 << 27)
#define SW1 (1 << 11)
#define SW2 (1 << 12)
#define ALL_LED (0xFF << 19)

void delay(uint32_t ms){
uint32_t i, j;
   for(i = 0; i < ms; i++){
      for(j = 0; j < 1250; j++){}
}
}

int main(){
   LPC_GPIO1 -> FIODIR |= BUZZER;
   LPC_GPIO1 -> FIODIR |= ALL_LED;
```

```
    LPC_GPIO2 -> FIODIR &= ~SW1;
    LPC_GPIO2 -> FIODIR &= ~SW2;
    LPC_GPIO1 -> FIOCLR = ALL_LED;
    LPC_GPIO1 -> FIOCLR = BUZZER;

    while(1){
        if ((LPC_GPIO2 -> FIOPIN & SW1) != 0){
            LPC_GPIO1 -> FIOSET = BUZZER;
        }

        else{
            LPC_GPIO1 -> FIOCLR = BUZZER;
        }

        if ((LPC_GPIO2 -> FIOPIN & SW2) != 0){
            LPC_GPIO1 -> FIOSET = ALL_LED;
        }

        else{
            LPC_GPIO1 -> FIOCLR = ALL_LED;
        }
    }
}
```

ss