Experiment No.1

```
// Roll no.304C068
// Name- Sanket Adsule



#include<reg51.h>
#define Del 3000
sfr LED_PORT2=0xA0;             // defining LED_PORT2 for PORT2

void delay(unsigned int x)              // delay function
{
        unsigned int i,j;
        for(i=0;i<=x;i++)

        for(j=0;j<=100;j++);



}

void main(void)
{
        while(1)                        // do it continuosly
                {
                        LED_PORT2=0xff;                 // LED ON
                        delay(Del);
                        LED_PORT2=0x00;         // LED OFF
                        delay(Del);
                }
}




//BCD Counter on PORT 1
BCD COUNTER
```

```
#include<reg51.h>
#define Del 2000


void delay(unsigned int x)          // delay function
{
    unsigned int i,j;
    for(i=0;i<x;i++)
            for(j=0;j<=100;j++);
}

void main(void)
{
unsigned char
count[10]={0xff,0xfe,0xfd,0xfc,0xfb,0xfa,0xf9,0xf8,0xf7,0xf6};

unsigned int x;

    P1=0x00;        // Make P1 as output port

    while(1)    // do it continuosly
        {
            for(x=0;x<10;x++)
                {
                    P1=count[x];
                    delay(Del);
                }
        }
}
```

Experiment no: 2

```
// Roll no.304C068
// Name- Sanket Adsule
//Sevan segment display

/HEX Counter on PORT 1
#include<reg51.h>
#define Del 2000

void delay(unsigned int x)            // delay function
{
    unsigned int i,j;
    for(i=0;i<x;i++)
            for(j=0;j<=100;j++);
}

void main(void)
{
    unsigned char
count[16]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,
0x83,0xc6,0xa1,0x86,0x8e};
    unsigned int x;
    P1=0x00;                                                      //
Make P1 as output port

    while(1)                                                      // do
it continuosly
        {
            for(x=0;x<16;x++)
                {
                    P1=count[x];
                    delay(Del);
                }
        }
```

Experiment No.3

```
// Roll no.304C068
// Name- Sanket Adsule

// Square waveform generation
#include<reg51.h>

void delay()
{
        inti,j;
        for(i=0;i<100;i++)
            for(j=0;j<100;j++);
}

void main()
{

while(1)
    {
P2=0x00;                // logic0 of sqaure wave
    delay();
        P2=0x0ff;               // logic 1 of sqaure wave
    delay();
    }
}
```

```
// Triangular waveform generation
#include< reg51.h>

unsigned char d;
void main(void)
{
while(1)
    {
for(d=0; d<255; d++)
        {           P2 = d;
        }
for(d=255; d>0; d--)
        {
            P2 = d;
        }
    }
}
```

Experiment No.4

```
// Roll no.304C068
// Name- Sanket Adsule


// Stepper clockwise

#include<reg51.h>

void delay()
{
        inti,j;
        for(i=0;i<2000;i++)
            for(j=0;j<100;j++);
}
void main()
{
    while(1)
    {
        P2=0x03;
        delay();
        P2=0x06;
        delay();
        P2=0x0C;
        delay();
        P2=0x09;
        delay();
    }
}
```

```
// Stepper Anticlockwise


#include<reg51.h>

void delay()
{
        inti,j;
        for(i=0;i<2000;i++)
            for(j=0;j<100;j++);
}

void main()
{
    while(1)
    {
        P2=0x09;
        delay();
        P2=0x0c;
        delay();
        P2=0x06;
        delay();
        P2=0x03;
        delay();
    }
}
```

```
// Stepper 8 step clockwise

#include<reg51.h>

void delay()
{
                inti,j;
                for(i=0;i<1000;i++)
                        for(j=0;j<100;j++);
}
void main()
{



while(1)
        {
                P2=0x01;
                delay();
                P2=0x03;
                delay();
                P2=0x02;
                delay();
                P2=0x06;
                delay();
                P2=0x04;
                delay();
                P2=0x0C;
                delay();
                P2=0x08;
                delay();
        P2=0x09;
        delay();

    }
}
```

Experiment No.5

```
// Roll no.304C068
// Name- Sanket Adsule
```

RAM-RAM

```
        ORG 0000H

        MOV R0, #20H
        MOV R1, #40H
        MOV R2, #05H

BACK:   MOV A, @R0
        MOV @R1, A
        INC R0
        INC R1
        DJNZ R2, BACK
        END
```

RAM-EX

```
    ORG 0000H

        MOV R0, #20H
        MOV DPTR, #040H
        MOV R2, #05H
```

```
BACK:     MOV A, @R0
          MOVX @DPTR, A
          INC R0
          INC DPTR
          DJNZ R2, BACK
          END
```

ROM-RAM

```
     ORG 0000H

          MOV R0, #20H
          MOV R1, #05H
          MOV DPTR, #0234H

BACK:     CLR A
          MOVC A, @A+DPTR
          MOV @R0, A
          INC R0
          INC DPTR
          DJNZ R1, BACK

          ORG 0234H
          DB 2H, 4H, 6H, 8H, 10H
          END
```