WHITE LINE FOLLOWER :

```
#include<avr/io.h>

void main()

{
        DDRD=0b11111111; //set PD4 as output bit

        DDRC=0b0000000;   //set PORTC as input port

        DDRB=0b00011110;          //PB1, PB2, PB3, PB4 as output port

    int ls=0, rs=0;       // define & initialize ls, rs integer as 0 to

                                        // acquire the left sensor status in ls and right sensor

                                        // status in rs


        while(1)          // create infinite loop

        {
        rs=(PINC&0b0000001);   //acquire only left sensor status connected at PC0

        ls=(PINC&0b0001000);   // acquire only right sensor status connected at PC3

        PORTD &= ~(1<<4);

                if((rs==0b0000000)&(ls==0b0000000)) //check sensor status for both sensor OFF

                {
                        PORTB=0b00010010;  //forward


                        ls=0;           //set sensor status off

                        rs=0;           //set sensor status off

                }


                if((rs==0b0000001)&(ls==0b0000000)) //check sensor status for left sensor=OFF
and
```

```c
                                        // right sensor=ON
            {

                    PORTB=0b00000010;    //turn left
                    PORTD = (1<<4);
                    ls=0;            //set sensor status off
                    rs=0;                            //set sensor status off
            }


            if((rs==0b0000000)&(ls==0b0001000)) //check sensor status for left sensor=ON
    and
                                    // right sensor=OFF
            {
                    PORTB=0b00010000;    //turn right
                    PORTD = (1<<4);
                    ls=0;            //set sensor status off
                    rs=0;                            ///set sensor status off
            }

    if((rs==0b0000001)&(ls==0b0001000)) //check sensor status for both sensor ON
            {
                    PORTB=0b00000000;  //stop
                    ls=0;            //set sensor status off
                    rs=0;            //set sensor status off


            }
        }
    }
```

RUNNING LEDS :

```c
#define F_CPU 120000000UL
#include<avr/io.h>
#include<util/delay.h>
#include "robosapiens.c"


int main(void)
{
DDRD=0b00010000;
DDRB=0b00011110;  // PB1,PB2,PB3 and PB4 of PORTB are set as output.


while(1)        // infinite while loop
  {
                for(int i=1; i<=4; i++)
                {
        PORTB = (1<<i);   // bitwise operation to glow leds after every 0.5 secs one by one
                PORTD=(1<<4);
                wait(0.5);       // wait function defined in robosapiens.c file function argument:
time in seconds
        PORTD = ~PORTD;
                wait(0.5);
          }
        }
}
```

OBSTACLE AVOIDER :

```c
#define F_CPU 12000000UL

#include<avr/io.h>

#include<util/delay.h>

#include"robosapiens.c"


int main(void)

{
        DDRD=0b11111111; //set PD4 as output bit

        DDRC=0b0000000;   //set PORTC as input port

        DDRB=0b00011110;          //PB1, PB2, PB3, PB4 as output port

    int ls=0, rs=0;       // define & initialize ls, rs integer as 0 to

                                        // acquire the left sensor status in ls and right sensor

                                        // status in rs


        while(1)        // create infinite loop
        {
        rs=(PINC&0b0000001);   //acquire only left sensor status connected at PC0
        ls=(PINC&0b0001000);   // acquire only right sensor status connected at PC3
        PORTD = ~PORTD;


                if((rs==0b0000001)||(ls==0b0001000))
                {
                PORTD=(1<<4);
                }
```

```c
            if((rs==0b0000000)&&(ls==0b0000000)) //check sensor status for both sensor
OFF

            {

                    PORTB=0b00010010;  //forward

                    ls=0;           //set sensor status off

                    rs=0;           //set sensor status off


            }


            if((rs==0b0000001)&&(ls==0b0000000))  //check sensor status for left
sensor=ON and

                                  // right sensor=OFF

            {

                    PORTD = (1<<4);

                    PORTB=0b00001100;    //backward

                    wait(0.5);

                    PORTB=0b00000010;    //turn left to avoid the obstacle

                    wait(1);

                    ls=0;           //set sensor status off

                    rs=0;                        //set sensor status off


            }



            if((rs==0b0000000)&&(ls==0b0001000))  //check sensor status for left
sensor=OFF and
```

```
                                        // right sensor=ON
            {

                    PORTD = (1<<4);
                    PORTB=0b00001100;    //backward
                    wait(0.5);
                    PORTB=0b00010000;    //turn right to avoid the obstacle
                    wait(1);
                    ls=0;            //set sensor status off
                    rs=0;                        //set sensor status off

            }



    if((rs==0b0000001)&&(ls==0b0001000)) //check sensor status for both sensor ON
            {
                    PORTB=0b00000000;  //stop
                    ls=0;          //set sensor status off
                    rs=0;          //set sensor status off
            }

        }
}
ROBOSAPIENS.C : void wait(float sec);

void wait(float sec)

{    for(int i=0;i<(int)46*sec;i++) _delay_loop_2(0);  }
```

LIGHT SEARCHING ROBOT :

```c
#include<avr/io.h>

int main(void)

{

DDRB=0b11111110; // PORTB as output Port connected to motors and PB0 as input port
connected to sensor


int light_sensor=0;


while(1) // infinite loop

{

    light_sensor=PINB&0b00000001; // mask PB0 bit of Port B


        if(light_sensor==0b00000001) //if Light source is on robot's left side "

    PORTB=0b00000001; // move left


        else


        PORTB=0b00001000; // move right


 }//while closed


}//main closed
```

START ON CLAP :

```c
#include<avr/io.h>
#include<util/delay.h>
#include "robosapiens.c"
int main(void)
{

      DDRB=0b00011110;  //PB1, PB2, PB3, PB4 as output port
   int ss=0;       // define & initialize ss integer as  0 to acquire the sound sensor status in ss


      while(1)       // create infinite loop
      {
      ss=(PINB&0b0000001);     //acquire only sound sensor status connected at PB0

            if(ss==0b0000001)     //check for sound sensor ON status at PB0
            {


         PORTB=0b00010010;   //forward
            wait(5);
         PORTB=0b00000000;       //stop


       }


      }

}
```