NAME: SADAR SARMAD.
ROLL_NUMBER: 22P-9009.
SECTION: BS_AI_4A.

CODE:

```
1.      .INCLUDE " M328pDEF.INC "
2.      .ORG 0

3.      LDI R21 , HIGH ( RAMEND )
4.      OUT SPH , R21
5.      LDI R21 , LOW ( RAMEND )
6.      OUT SPL , R21

7.      LDI R31 , 0xFF
8.      OUT DDRD , R31
9.      OUT DDRB , R31
10.     CBI PORTB , 2
11.     CALL DELAY_2ms
12.     LDI R30 , 0x38 ;; Function set : 8 -bit , 2 lines , 5x8 dots
13.     CALL CMNDWRT
14.     CALL DELAY_2ms
15.     LDI R30 , 0x0E ;; Display on , cursor on , blink off
16.     CALL CMNDWRT
17.     CALL DELAY_2ms
18.     LDI R30 , 0x01 ;; Clear display
19.     CALL CMNDWRT
20.     CALL DELAY_2ms
21.     LDI R30 , 0x06 ;; Entry mode set : increment cursor , no shift
22.     CALL CMNDWRT
23.     CALL DELAY_2ms
24.     LDI R30 , 'C'
25.     CALL DATAWRT
26.     LDI R30 , 'O'
27.     CALL DATAWRT
28.     LDI R30 , 'A'
29.     CALL DATAWRT
30.     LDI R30 , 'L'
31.     CALL DATAWRT

32.     HERE : JMP HERE

33.     CMNDWRT :
34.     OUT PORTD , R30
35.     CBI PORTB , 0
36.     CBI PORTB , 1
37.     SBI PORTB , 2
38.     CALL DELAY_100us
39.     CBI PORTB , 2
40.     CALL DELAY_100us
41.     RET

42.     DATAWRT :
43.     OUT PORTD , R30
44.     SBI PORTB , 0
45.     CBI PORTB , 1
46.     SBI PORTB , 2
47.     CALL DELAY_100us
48.     CBI PORTB , 2
49.     CALL DELAY_100us
50.     RET

51.     // Delays ---------------------------------

52.     SDELAY :
53.     NOP
54.     NOP
55.     RET

56.     DELAY_100us :
57.     PUSH R29
58.     LDI R17 , 60
59.     DRO : CALL SDELAY
60.     DEC R29
61.     BRNE DRO
62.     POP R29
```

```
63.    RET

64.    DELAY_2ms :
65.    PUSH R29
66.    LDI R29 , 20
67.    LDR0 : CALL DELAY_100us
68.    DEC R29
69.    BRNE LDR0
70.    POP R29
71.    RET
```

● Initialization and Setup

● Include Device-Specific Definitions and Set Origin:

1. The code includes the device-specific definitions for the ATmega328p microcontroller, allowing it to use predefined register names and constants.
2. It sets the starting address of the program to 0.

● Initialize Stack Pointer:

1. The stack pointer is set to the highest address in the RAM. This is necessary for proper stack operation, which is used for subroutine calls and local variable storage.

● Configure Data Direction Registers:

1. All pins of Port D and Port B are configured as output. This is done by writing `0xFF` (all bits set) to `DDRD` and `DDRB`.

● Prepare for LCD Communication:

1. The bit 2 of `PORTB` is cleared to prepare the LCD for communication.
2. A delay subroutine (`DELAY_2ms`) is called to wait for 2 milliseconds.
3. LCD Initialization

● Function Set:

1. The LCD is configured for 8-bit mode, 2-line display, and 5x8 dot character font by writing `0x38` to the command register of the LCD.

● Display Control:

1. The display is turned on with the cursor on and blink off by writing `0x0E` to the command register.

● Clear Display:

1. The display is cleared by writing `0x01` to the command register.

● Entry Mode Set:

1. The entry mode is set to increment the cursor position and not shift the display by writing `0x06` to the command register.

● Display "COAL"

● Write Characters to Display:

1. The characters 'C', 'O', 'A', and 'L' are written to the LCD data register one by one. This displays "COAL" on the LCD screen.

● Infinite Loop

1. **Halt the Program:**
2. The program enters an infinite loop, halting any further execution.

● Subroutines

● Command Write Subroutine (CMNDWRT):

1. Sends a command to the LCD. The command is placed on `PORTD`, and specific control signals are toggled on `PORTB` to signal the LCD to read the command.

● Data Write Subroutine (DATAWRT):

1. Sends data to the LCD. The data is placed on `PORTD`, and control signals are toggled on `PORTB` to signal the LCD to read the data.

● Delay Subroutines:

1. `SDELAY` introduces a very short delay using `NOP` (no operation) instructions.
2. `DELAY_100us` introduces a delay of approximately 100 microseconds by calling `SDELAY` multiple times.
3. `DELAY_2ms` introduces a delay of approximately 2 milliseconds by calling `DELAY_100us` multiple times.