

```

#include "p16F887.inc"

; CONFIG1
; __config 0x23E4
__CONFIG __CONFIG1, _FOSC_INTRC_NOCLKOUT & _WDTE_OFF & _PWRTE_ON & _MCLRE_ON &
        _CP_OFF & _CPD_OFF & _BOREN_ON & _IESO_OFF & _FCMEN_OFF & _LVP_OFF
; CONFIG2
; __config 0x3FFF
__CONFIG __CONFIG2, _BOR4V_BOR40V & _WRT_OFF

ESTADO_ABC EQU 0x20
;Este registro de proposito genral se va a utilizar para ver el estado del
; abecedario que se va a desplegar, por ejemplo en la posicion inicial su
;valor sera 1 y inidica que se esta seleccionando la letra a, por ejemplo si
;fuera este valor igual a 2 indicaria que se esta seleccionando la letra b
INTENTO EQU 0x21; Guarda la cantidad de errores que puede cometer al adivinar
RS EQU 0x03;Hace referencia al pin de RS de la pantalla LCD
E EQU 0x02;Hace referencia al pin de RS de la pantalla LCD
ANT_ESTADO_ABC EQU 0x22; Guarda un valor se utliza para no entrar a un ciclo
;infinito
;POSX se guardan el valor de la letra a adivinar
POS1 EQU 0x23
POS2 EQU 0x24
POS3 EQU 0x25
POS4 EQU 0x26
POS5 EQU 0x27
POS6 EQU 0x28
POS7 EQU 0x29
POS8 EQU 0x2A
POS9 EQU 0x2B
POS10 EQU 0x2C
POS11 EQU 0x2D
POS12 EQU 0x2E
POS13 EQU 0x2F
POS14 EQU 0x30
POS15 EQU 0x31
POS16 EQU 0x32
TOTALDELETRAS EQU 0x33;longitud de la palabra
INSERTAOADIVINA EQU 0x34;// Estado donde mencina si se esta introduciendo la
;palabra o se esta adiviando la palabra
CHECAPOST EQU 0x34;Variable que tomas los valores de POS1 se utiliza para entrar
; en un ciclo por lo que se guarda un valor POSX
COMP EQU 0x36;Variable que tomas los valores se utiliza para entrar en un ciclo
ACIERTA EQU 0x37; ES 1 si acierta
; Registros valen 1 si ya se adivino la letra de la palabra
EPOS1 EQU 0x38

```

```
EPOS2 EQU 0x39
EPOS3 EQU 0x3A
EPOS4 EQU 0x3B
EPOS5 EQU 0x3C
EPOS6 EQU 0x3D
EPOS7 EQU 0x3E
EPOS8 EQU 0x3F
EPOS9 EQU 0x40
EPOS10 EQU 0x41
EPOS11 EQU 0x42
EPOS12 EQU 0x43
EPOS13 EQU 0x44
EPOS14 EQU 0x45
EPOS15 EQU 0x46
EPOS16 EQU 0x47
ADIVINATOTAL EQU 0x48; si adivina es diferente a 0
ORG 0; Inicia programa en 0 donde es el stack pointer
goto POGRAMA
ORG 4; Interrupcion
goto INTER

INTER; Comienzan las interrupciones
btfsc PORTB,7; se presiono Izquierda?
call IZQUIERDA
btfsc PORTB,6; se presiono Derecha?
call DERECHA
btfsc PORTB,5; se presiono seleccion?
call SELECCION
btfsc PORTB,4; se presiono cambio de modo?
call ADIVINAR
bsf INTCON, RBIE; se activa interrupcion extena por los pines del puertob
bcf INTCON, RBIF; se apaga bandera
retfie; regreso al programa principal

IZQUIERDA
; Si es 1 se cambia a 26 de A a Z
; Si no se resta 1 al estado del Abecedario
bcf INTCON, RBIF
movlw D'1'
subwf ESTADO_ABC,0
btfss STATUS,Z
goto ESTADOMENOS1
bcf STATUS,Z
movlw D'26'
movwf ESTADO_ABC
return
ESTADOMENOS1
```

```
movlw D'1'
subwf ESTADO_ABC,1
return

DERECHA
; Si es 26 se cambia a 1 de Z a A
; Si no se suma 1 al estado del Abecedario
bcf INTCON, RBIF
movlw D'26'
subwf ESTADO_ABC,0
btfss STATUS,Z
goto ESTADOMAS1
bcf STATUS,Z
movlw D'1'
movwf ESTADO_ABC
return
ESTADOMAS1
movlw D'1'
addwf ESTADO_ABC,1
return

SELECCION
; Hay dos modos de seleccion cuando se selecciona la palabra o cuando se adivina
bcf STATUS,Z
movlw D'1';
subwf INSERTAOADIVINA,0
btfss STATUS,Z
goto ADELANTE
clrf ADIVINATOTAL
bcf STATUS,Z
goto SEGUNDAADIVINAR; va a imprimir la segunda linea y verificar si acerto
REGRESO5
clrf ANT_ESTADO_ABC

bcf STATUS,Z
movlw D'1';
subwf ADIVINATOTAL,0; Si adivina sus intentos quedan como estan
btfsc STATUS,Z
return

bcf STATUS,Z
movlw D'1';
subwf INTENTO,1 ; si no adivina se le quita un intento
btfss STATUS,Z
return
```

```
bcf STATUS,Z
movlw D'0';
subwf INTENTO,0; si los intestos son ceros imprime que ha perdido
btfss STATUS,Z
return

bcf PORTB,RS
movlw b'10000000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
call lh
call la
call ls
call lvacio
call lp
call le
call lr
call ld
call li
call ld
call lo
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
    fin
    goto fin

    return

ADELANTE; En modo insertar palabra
; se guarda letra en el registro correcto
bcf INTCON, RBIF
movlw D'1'
addwf TOTALDELETRAS,1

movlw D'1';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN1

movlw D'2';
```

```
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN2

movlw D'3';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN3

movlw D'4';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN4

movlw D'5';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN5

movlw D'6';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN6

movlw D'7';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN7

movlw D'8';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN8

movlw D'9';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN9

movlw D'10';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN10

movlw D'11';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
```

```
goto GUARDAREN11

movlw D'12';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN12

movlw D'13';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN13

movlw D'14';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN14

movlw D'15';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN15

movlw D'16';
subwf TOTALDELETRAS,0
btfsc STATUS,Z
goto GUARDAREN16

DESPUESDEGUARDAR
return

GUARDAREN1
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS1
goto DESPUESDEGUARDAR

GUARDAREN2
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS2
goto DESPUESDEGUARDAR

GUARDAREN3
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS3
goto DESPUESDEGUARDAR
```

```
GUARDARENP4
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS4
goto DESPUESDEGUARDAR
```

```
GUARDARENP5
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS5
goto DESPUESDEGUARDAR
```

```
GUARDARENP6
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS6
goto DESPUESDEGUARDAR
```

```
GUARDARENP7
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS7
goto DESPUESDEGUARDAR
```

```
GUARDARENP8
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS8
goto DESPUESDEGUARDAR
```

```
GUARDARENP9
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS9
goto DESPUESDEGUARDAR
```

```
GUARDARENP10
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS10
goto DESPUESDEGUARDAR
```

```
GUARDARENP11
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS11
```

```
goto DESPUESDEGUARDAR

GUARDAREN12
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS12
goto DESPUESDEGUARDAR

GUARDAREN13
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS13
goto DESPUESDEGUARDAR

GUARDAREN14
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS14
goto DESPUESDEGUARDAR

GUARDAREN15
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS15
goto DESPUESDEGUARDAR

GUARDAREN16
bcf STATUS,Z
movfw ESTADO_ABC
movwf POS16
goto ADIVINAR

ADIVINAR; Se activa modo adivinar
bcf INTCON, RBIF
movlw d'1'
movwf INSERTAOADIVINA
bcf STATUS,RP1
bsf STATUS,RP0;Banco 1
movlw b'11100000'
movwf IOCB; Registro de eleccion de la interrupcion por cambio de nivel del puerto B
;se desactiva la interrupcion por ese pin
bcf STATUS,RP1
bcf STATUS,RP0;Banco 1
goto borrarletras; borra las letras y pone guiones
despuesdeborrarletras
bcf STATUS,Z
return
```



```
borrarletras; borra las letras y pone guiones
bcf PORTB,RS
movlw b'11000000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO

bcf STATUS,Z

movlw D'0';
subwf POS1,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS2,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS3,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS4,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS5,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS6,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS7,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
```

```
movlw D'0';
subwf POS8,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS9,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS10,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS11,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS12,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS13,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS14,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS15,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
movlw D'0';
subwf POS16,0
btfsc STATUS,Z
goto despuesdeborrarletras
call guionbajo
goto despuesdeborrarletras
```

```
POGRAMA; incio de pograma
bcf STATUS,RP1
bsf STATUS,RP0;Banco 1
clrf TRISA
clrf TRISC
clrf TRISD
clrf TRISE; como salidas
movlw b'11110000'
movwf TRISB
movlw b'11110000'
movwf IOCB; Registro de eleccion de la interrucion por cambio de nivel del puerto B
movlw b'10000010'
movwf OPTION_REG; para futuros retardos
bsf STATUS,RP1
bsf STATUS,RP0;Banco 3
clrf ANSELH
clrf ANSEL; entradas digitales
bcf STATUS,RP1
bcf STATUS,RP0;Banco 0
clrf PORTA
clrf PORTC
clrf PORTD
clrf PORTE
clrf PORTB; se limpian puertos
movlw D'7'; total de intentos
movwf INTENTO

clrf TOTALDELETRAS
clrf POS1
clrf POS2
clrf POS3
clrf POS4
clrf POS5
clrf POS6
clrf POS7
clrf POS8
clrf POS9
clrf POS10
clrf POS11
clrf POS12
clrf POS13
clrf POS14
clrf POS15
clrf POS16
```

```
clrf INSERTAOADIVINA
clrf EPOS1
clrf EPOS2
clrf EPOS3
clrf EPOS4
clrf EPOS5
clrf EPOS6
clrf EPOS7
clrf EPOS8
clrf EPOS9
clrf EPOS10
clrf EPOS11
clrf EPOS12
clrf EPOS13
clrf EPOS14
clrf EPOS15
clrf EPOS16 ;se liampian registros
movlw b'00111000';////////Prender LCD
movwf PORTD
bsf PORTB, E
bcf PORTB, E
call RETARDO
movlw b'00000110';////////Prender LCD
movwf PORTD
bsf PORTB, E
bcf PORTB, E
call RETARDO
movlw b'00001111';////////Prender LCD
movwf PORTD
bsf PORTB, E
bcf PORTB, E
call RETARDO
bsf PORTB, RS

bsf INTCON, GIE; se activa interrupcion global
bsf INTCON, RBIE; se activa interrupcion ext por el puerto B
bcf INTCON, RBIF; se liampia bandera

movlw D'1'; el pograma empieza apunta a la letra a
movwf ESTADO_ABC
movlw D'0'
movwf ANT_ESTADO_ABC

espera; parte de espera de la interrupcion
call RETARDOLARGO
bcf INTCON, RBIF
bsf INTCON, RBIE
```

```
BSF INTCON, GIE
movfw ANT_ESTADO_ABC
subwf ESTADO_ABC,0
btfss STATUS,Z
goto COMPARA
bcf STATUS,Z
goto espera

COMPARA; compara el estado de la posiscion y imprime el abecedario correctamente
movfw ESTADO_ABC
movwf ANT_ESTADO_ABC

movlw D'1';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pa

movlw D'2';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pb

movlw D'3';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pc

movlw D'4';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pd

movlw D'5';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pe

movlw D'6';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pf

movlw D'7';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pg
```

```
movlw D'8';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto ph

movlw D'9';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pi

movlw D'10';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pj

movlw D'11';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pk

movlw D'12';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pl

movlw D'13';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pm

movlw D'14';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pn

movlw D'15';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto po

movlw D'16';
subwf ESTADO_ABC,0
btfsc STATUS,Z
goto pp

movlw D'17';
subwf ESTADO_ABC,0
```

```
btfs STATUS,Z
goto pq

movlw D'18';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pr

movlw D'19';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto ps

movlw D'20';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pt

movlw D'21';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pu

movlw D'22';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pv

movlw D'23';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pw

movlw D'24';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto px

movlw D'25';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto py

movlw D'26';
subwf ESTADO_ABC,0
btfs STATUS,Z
goto pz
```

```
;Si el estado es 1 imprime e la primera linea apuntando a la letra a
; y tambien el numero de intentos
pa
call borrarlcd
bsf INTCON, RBIE
bcf STATUS,Z
call numintentos
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call lder
call la
call lizq
call lb
call lc
call ld
call le
call lf
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pb
call borrarlcd
bsf INTCON, RBIE
bcf STATUS,Z
call numintentos
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call la
call lder
call lb
call lizq
call lc
call ld
call le
call lf
call lg
```



```
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pc
call borrarlcd
bsf INTCON, RBIE
bcf STATUS, Z
call numintentos
call lvacio
call lvacio
call lvacio
call lvacio
call la
call lb
call lder
call lc
call lizq
call ld
call le
call lf
call lg
call lh
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pd
call borrarlcd
bsf INTCON, RBIE
bcf STATUS, Z
call numintentos
call lvacio
call lvacio
call lvacio
call la
call lb
call lc
call lder
call ld
call lizq
call le
call lf
call lg
call lh
call li
call lvacio
```

```
call IMPRIMIRSEGUNDALINEA
goto espera

pe
call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lvacio
call la
call lb
call lc
call ld
call lder
call le
call lizq
call lf
call lg
call lh
call li
call lj
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pf
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call la
call lb
call lc
call ld
call le
call lder
call lf
call lizq
call lg
call lh
call li
call lj
call lk
call lvacio
call IMPRIMIRSEGUNDALINEA
```

```
goto espera

pg
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lb
call lc
call ld
call le
call lf
call lder
call lg
call lizq
call lh
call li
call lj
call lk
call ll
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

ph
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lc
call ld
call le
call lf
call lg
call lder
call lh
call lizq
call li
call lj
call lk
call ll
call lm
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pi
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call ld
call le
call lf
call lg
call lh
call lder
call li
call lizq
call lj
call lk
call ll
call lm
call ln
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pj
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call le
call lf
call lg
call lh
call li
call lder
call lj
call lizq
call lk
call ll
call lm
call ln
call lo
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pk
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lf
call lg
call lh
call li
call lj
call lder
call lk
call lizq
call ll
call lm
call ln
call lo
call lp
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pl
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lg
call lh
call li
call lj
call lk
call lder
call ll
call lizq
call lm
call ln
call lo
call lp
call lq
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pm
```

```
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lh
call li
call lj
call lk
call ll
call lder
call lm
call lizq
call ln
call lo
call lp
call lq
call lr
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pn
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call li
call lj
call lk
call ll
call lm
call lder
call ln
call lizq
call lo
call lp
call lq
call lr
call ls
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

po
    call borrarlcd
```

```
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lj
call lk
call ll
call lm
call ln
call lder
call lo
call lizq
call lp
call lq
call lr
call ls
call lt
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pp
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lk
call ll
call lm
call ln
call lo
call lder
call lp
call lizq
call lq
call lr
call ls
call lt
call lu
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pq
    call borrarlcd
    bsf INTCON, RBIE
```

```
    bcf STATUS,Z
call numintentos
call lvacio
call ll
call lm
call ln
call lo
call lp
call lder
call lq
call lizq
call lr
call ls
call lt
call lu
call lv
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pr
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lm
call ln
call lo
call lp
call lq
call lder
call lr
call lizq
call ls
call lt
call lu
call lv
call lw
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

ps
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
```



```
call numintentos
call lvacio
call ln
call lo
call lp
call lq
call lr
call lder
call ls
call lizq
call lt
call lu
call lv
call lw
call lx
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pt
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lo
call lp
call lq
call lr
call ls
call lder
call lt
call lizq
call lu
call lv
call lw
call lx
call ly
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
pu
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
```

```
call lvacio
call lp
call lq
call lr
call ls
call lt
call lder
call lu
call lizq
call lv
call lw
call lx
call ly
call lz
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pv
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lq
call lr
call ls
call lt
call lu
call lder
call lv
call lizq
call lw
call lx
call ly
call lz
call lvacio
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pw
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
```

```
call lr
call ls
call lt
call lu
call lv
call lder
call lw
call lizq
call lx
call ly
call lz
call lvacio
call lvacio
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
px
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call ls
call lt
call lu
call lv
call lw
call lder
call lx
call lizq
call ly
call lz
call lvacio
call lvacio
call lvacio
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera
```

```
py
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lt
```

```
call lu
call lv
call lw
call lx
call lder
call ly
call lizq
call lz
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

pz
    call borrarlcd
    bsf INTCON, RBIE
    bcf STATUS,Z
call numintentos
call lvacio
call lu
call lv
call lw
call lx
call ly
call lder
call lz
call lizq
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call lvacio
call IMPRIMIRSEGUNDALINEA
goto espera

; Instrucciones para imprimir X letra o caracter
la
bsf PORTB,RS
movlw "A"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
```

```
call RETARDO
return

lb
bsf PORTB,RS
movlw "B"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

lc
bsf PORTB,RS
movlw "C"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

ld
bsf PORTB,RS
movlw "D"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

le
bsf PORTB,RS
movlw "E"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

lf
bsf PORTB,RS
movlw "F"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lg
bsf PORTB,RS
movlw "G"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lh
bsf PORTB,RS
movlw "H"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
li
bsf PORTB,RS
movlw "I"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lj
bsf PORTB,RS
movlw "J"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lk
bsf PORTB,RS
movlw "K"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
ll
```

```
bsf PORTB,RS
movlw "L"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lm
bsf PORTB,RS
movlw "M"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
ln
bsf PORTB,RS
movlw "N"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lo
bsf PORTB,RS
movlw "O"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lp
bsf PORTB,RS
movlw "P"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lq
bsf PORTB,RS
movlw "Q"
```

```
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lr
bsf PORTB,RS
movlw "R"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
ls
bsf PORTB,RS
movlw "S"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lt
bsf PORTB,RS
movlw "T"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lu
bsf PORTB,RS
movlw "U"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lv
bsf PORTB,RS
movlw "V"
movwf PORTD
bsf PORTB,E
```



```
bcf PORTB,E
call RETARDO
return
```

```
lw
bsf PORTB,RS
movlw "W"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lx
bsf PORTB,RS
movlw "X"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
ly
bsf PORTB,RS
movlw "Y"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
lz
bsf PORTB,RS
movlw "Z"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return
```

```
l1
bsf PORTB,RS
movlw "1"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
```

return

l2

bsf PORTB,RS

movlw "2"

movwf PORTD

bsf PORTB,E

bcf PORTB,E

call RETARDO

return

l3

bsf PORTB,RS

movlw "3"

movwf PORTD

bsf PORTB,E

bcf PORTB,E

call RETARDO

return

l4

bsf PORTB,RS

movlw "4"

movwf PORTD

bsf PORTB,E

bcf PORTB,E

call RETARDO

return

l5

bsf PORTB,RS

movlw "5"

movwf PORTD

bsf PORTB,E

bcf PORTB,E

call RETARDO

return

l6

bsf PORTB,RS

movlw "6"

movwf PORTD

bsf PORTB,E

bcf PORTB,E

call RETARDO

return

```
l7
bsf PORTB,RS
movlw "7"
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

lvacio
bsf PORTB,RS
movlw b'00100000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

lizq
bsf PORTB,RS
movlw b'01111111'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

lder
bsf PORTB,RS
movlw b'01111110'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

guionbajo
bsf PORTB,RS
movlw b'01011111'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
call RETARDO
return

RETARDO; retardo que necesita la lcd para imprimir un caracter
```

```
call RETARDO10NOP
call RETARDO10NOP
call RETARDO10NOP
call RETARDO10NOP
return

numintentos; imprimir el numero de intentos
movlw D'1';
subwf INTENTO,0
btfsc STATUS,Z
goto l1
movlw D'2';
subwf INTENTO,0
btfsc STATUS,Z
goto l2
movlw D'3';
subwf INTENTO,0
btfsc STATUS,Z
goto l3
movlw D'4';
subwf INTENTO,0
btfsc STATUS,Z
goto l4
movlw D'5';
subwf INTENTO,0
btfsc STATUS,Z
goto l5
movlw D'6';
subwf INTENTO,0
btfsc STATUS,Z
goto l6
movlw D'7';
subwf INTENTO,0
btfsc STATUS,Z
goto l7
return

borrarlcd; aunque se llame borrar solo cambia el puntero a la primera posicion
bcf PORTB,RS
movlw b'10000000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
return
```

```
RETARDOLARGO; retardo que necesita la lcd para borrar LCD o cambiar puntero
bcf INTCON,2
bcf STATUS,RP1
bsf STATUS,RP0;Banco 1
movlw b'10000010'
movwf OPTION_REG
bcf STATUS,RP1
bcf STATUS,RP0;Banco 0
    movlw D'51'
    movwf TMR0
    bcf INTCON,2
    ciclo
    btfss INTCON,2
    goto ciclo
    bcf INTCON,2
    movlw D'51'
    movwf TMR0
return
```

RETARDO10NOP; retardo con nop

```
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
return
```

IMPRIMIRSEGUNDALINEA ; imprimie la segunda linea

```
movlw D'0';
subwf INSERTAOADIVINA,0
btfsc STATUS,Z
call SEGUNDAINSERTAR
return
```

SEGUNDAINSERTAR; imprime la segunda lineas si se esta insertando la palabra

```
bcf PORTB,RS
movlw b'11000000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
```

```
movfw POS1
movwf CHECAPOST
call CHECA
movfw POS2
movwf CHECAPOST
call CHECA
movfw POS3
movwf CHECAPOST
call CHECA
movfw POS4
movwf CHECAPOST
call CHECA
movfw POS5
movwf CHECAPOST
call CHECA
movfw POS6
movwf CHECAPOST
call CHECA
movfw POS7
movwf CHECAPOST
call CHECA
movfw POS8
movwf CHECAPOST
call CHECA
movfw POS9
movwf CHECAPOST
call CHECA
movfw POS10
movwf CHECAPOST
call CHECA
movfw POS11
movwf CHECAPOST
call CHECA
movfw POS12
movwf CHECAPOST
call CHECA
movfw POS13
movwf CHECAPOST
call CHECA
movfw POS14
movwf CHECAPOST
call CHECA
movfw POS15
movwf CHECAPOST
call CHECA
movfw POS16
movwf CHECAPOST
```

```
call CHECA
return

SEGUNDAADIVINAR; imprime la segunda lineas si se esta adivinando la palabra
; ademas checa si le atinaron a la letra
bcf STATUS,Z
movlw D'0'
subwf EPOS1,0
btfss STATUS,Z
goto S1
bcf PORTB,RS
movlw b'11000000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS1
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S1
movlw d'1'
movwf EPOS1
S1

bcf STATUS,Z
movlw D'0'
subwf EPOS2,0
btfss STATUS,Z
goto S2
bcf PORTB,RS
movlw b'11000001'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS2
movwf COMP
call SEGUNDAADIVINARCICLO
```

```
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S2
movlw d'1'
movwf EPOS1
S2

bcf STATUS,Z
movlw D'0'
subwf EPOS3,0
btfss STATUS,Z
goto S3
bcf PORTB,RS
movlw b'11000010'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS3
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S3
movlw d'1'
movwf EPOS1
S3

bcf STATUS,Z
movlw D'0'
subwf EPOS4,0
btfss STATUS,Z
goto S4
bcf PORTB,RS
movlw b'11000011'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS4
movwf COMP
```



```
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S4
movlw d'1'
movwf EPOS1
S4

bcf STATUS,Z
movlw D'0'
subwf EPOS5,0
btfss STATUS,Z
goto S5
bcf PORTB,RS
movlw b'11000100'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS5
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S5
movlw d'1'
movwf EPOS1
S5

bcf STATUS,Z
movlw D'0'
subwf EPOS6,0
btfss STATUS,Z
goto S6
bcf PORTB,RS
movlw b'11000101'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS6
```

```
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S6
movlw d'1'
movwf EPOS1
S6

bcf STATUS,Z
movlw D'0'
subwf EPOS7,0
btfss STATUS,Z
goto S7
bcf PORTB,RS
movlw b'11000110'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS7
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S7
movlw d'1'
movwf EPOS1
S7

bcf STATUS,Z
movlw D'0'
subwf EPOS8,0
btfss STATUS,Z
goto S8
bcf PORTB,RS
movlw b'11000111'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
```

```
movfw POS8
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S8
movlw d'1'
movwf EPOS1
S8

bcf STATUS,Z
movlw D'0'
subwf EPOS9,0
btfss STATUS,Z
goto S9
bcf PORTB,RS
movlw b'11001000'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS9
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S9
movlw d'1'
movwf EPOS1
S9

bcf STATUS,Z
movlw D'0'
subwf EPOS10,0
btfss STATUS,Z
goto S10
bcf PORTB,RS
movlw b'11001001'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
```

```
call RETARDOLARGO
movfw POS10
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S10
movlw d'1'
movwf EPOS1
S10

bcf STATUS,Z
movlw D'0'
subwf EPOS11,0
btfss STATUS,Z
goto S11
bcf PORTB,RS
movlw b'11001010'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS11
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S11
movlw d'1'
movwf EPOS1
S11

bcf STATUS,Z
movlw D'0'
subwf EPOS12,0
btfss STATUS,Z
goto S12
bcf PORTB,RS
movlw b'11001011'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
```

```
bsf PORTB,RS
call RETARDOLARGO
movfw POS12
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S12
movlw d'1'
movwf EPOS1
S12

bcf STATUS,Z
movlw D'0'
subwf EPOS13,0
btfss STATUS,Z
goto S13
bcf PORTB,RS
movlw b'11001100'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS13
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S13
movlw d'1'
movwf EPOS1
S13

bcf STATUS,Z
movlw D'0'
subwf EPOS14,0
btfss STATUS,Z
goto S14
bcf PORTB,RS
movlw b'11001101'
movwf PORTD
bsf PORTB,E
```

```
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS14
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S14
movlw d'1'
movwf EPOS1
S14

bcf STATUS,Z
movlw D'0'
subwf EPOS15,0
btfss STATUS,Z
goto S15
bcf PORTB,RS
movlw b'11001110'
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
movfw POS15
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S15
movlw d'1'
movwf EPOS1
S15

bcf STATUS,Z
movlw D'0'
subwf EPOS16,0
btfss STATUS,Z
goto S16
    bcf PORTB,RS
movlw b'11001111'
```

```
movwf PORTD
bsf PORTB,E
bcf PORTB,E
bsf PORTB,RS
call RETARDOLARGO
    movfw POS16
movwf COMP
call SEGUNDAADIVINARCICLO
bcf STATUS,Z
movlw D'0'
subwf ACIERTA,0
btfsc STATUS,Z
goto S16
movlw d'1'
movwf EPOS1
    S16
    bcf STATUS,Z
goto REGRESO5

SEGUNDAADIVINARCICLO; si adivina la letra la imprime
bcf STATUS,Z
movfw COMP
subwf ESTADO_ABC,0
btfss STATUS,Z
goto NOES
goto SIES

SIES
bcf STATUS,Z
movlw D'1'
movwf ACIERTA
movwf ADIVINATOTAL
movfw COMP
movwf CHECAPOST
call CHECA

NOES
return

CHECA;Esta subrutina imprime la letra que este contenida en CHECAPOST
    bcf STATUS,Z
    movlw D'0';
    subwf CHECAPOST,0
```

```
btfs STATUS,Z
call lvacio

bcf STATUS,Z
movlw D'1';
subwf CHECAPOST,0
btfs STATUS,Z
call la

bcf STATUS,Z
movlw D'2';
subwf CHECAPOST,0
btfs STATUS,Z
call lb

bcf STATUS,Z
movlw D'3';
subwf CHECAPOST,0
btfs STATUS,Z
call lc

bcf STATUS,Z
movlw D'4';
subwf CHECAPOST,0
btfs STATUS,Z
call ld

bcf STATUS,Z
movlw D'5';
subwf CHECAPOST,0
btfs STATUS,Z
call le

bcf STATUS,Z
movlw D'6';
subwf CHECAPOST,0
btfs STATUS,Z
call lf

bcf STATUS,Z
movlw D'7';
subwf CHECAPOST,0
btfs STATUS,Z
call lg

bcf STATUS,Z
movlw D'8';
```



```
subwf CHECAPOST,0
btfsc STATUS,Z
call lh

bcf STATUS,Z
movlw D'9';
subwf CHECAPOST,0
btfsc STATUS,Z
call li

bcf STATUS,Z
movlw D'10';
subwf CHECAPOST,0
btfsc STATUS,Z
call lj

bcf STATUS,Z
movlw D'11';
subwf CHECAPOST,0
btfsc STATUS,Z
call lk

bcf STATUS,Z
movlw D'12';
subwf CHECAPOST,0
btfsc STATUS,Z
call ll

bcf STATUS,Z
movlw D'13';
subwf CHECAPOST,0
btfsc STATUS,Z
call lm

bcf STATUS,Z
movlw D'14';
subwf CHECAPOST,0
btfsc STATUS,Z
call ln

bcf STATUS,Z
movlw D'15';
subwf CHECAPOST,0
btfsc STATUS,Z
call lo

bcf STATUS,Z
```

```
movlw D'16';
subwf CHECAPOST,0
btfsc STATUS,Z
call lp

bcf STATUS,Z
movlw D'17';
subwf CHECAPOST,0
btfsc STATUS,Z
call lq

bcf STATUS,Z
movlw D'18';
subwf CHECAPOST,0
btfsc STATUS,Z
call lr

bcf STATUS,Z
movlw D'19';
subwf CHECAPOST,0
btfsc STATUS,Z
call ls

bcf STATUS,Z
movlw D'20';
subwf CHECAPOST,0
btfsc STATUS,Z
call lt

bcf STATUS,Z
movlw D'21';
subwf CHECAPOST,0
btfsc STATUS,Z
call lu

bcf STATUS,Z
movlw D'22';
subwf CHECAPOST,0
btfsc STATUS,Z
call lv

bcf STATUS,Z
movlw D'23';
subwf CHECAPOST,0
btfsc STATUS,Z
call lw
```

```
bcf STATUS,Z
movlw D'24';
subwf CHECAPOST,0
btfsc STATUS,Z
call lx

bcf STATUS,Z
movlw D'25';
subwf CHECAPOST,0
btfsc STATUS,Z
call ly

bcf STATUS,Z
movlw D'26';
subwf CHECAPOST,0
btfsc STATUS,Z
call lz
bcf STATUS,Z
return

END;Fin del pograma
```