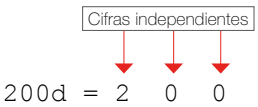
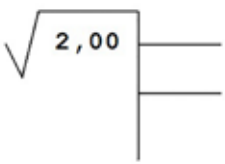
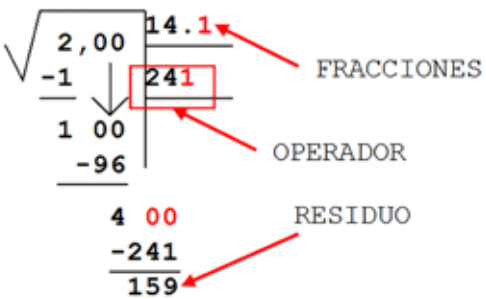
Encabezado para cualquier AVR

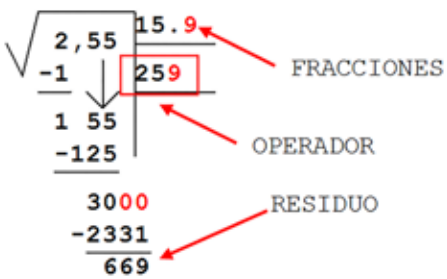
```
.DEF   ANS = R0     ;RESULTADO
.DEF     A = R16    ;RADICANDO
.DEF     B = R18    ;Para suma

LDI A,25            ;Aquí cargamos el RADICANDO
LDI B,0

SQRT:
LOOP:
SUB A,B             ;Substracción del RADICANDO
BRCS DONE           ;Si es más grande que el cuadrado
                    ;terminamos
CPI A,2
BRLO DONE
INC ANS             ;Incremento de RESULTADO
SUBI B,-2           ;Incremento de B por 2
RJMP LOOP

DONE:RJMP DONE
```






```
   OPER_0     OPER_1       OPER_2       ...ETC
;x      0     x     1      x     2
; ------     ------       ------
```


2,00
14.1
-1
241
1 00
-96
4 00
-241
159
FRACCIONES
OPERADOR
RESIDUO




```
;SUBRUTINA QUE EXTRAE LA RAÍZ CUADRADA DE UN NÚMERO DE
;8-Bits CON PUNTO DECIMAL
```

Encabezado para ATmega8515

Stack Pointer para ATmega8515

```
;HAREMOS LA SUBRUTINA DE RAÍZ CUADRADA SEGÚN LA SIGUIENTE
;ESTRUCTURA:
; ___________
;|RADICANDO  |
;           |RESULTADO
;           |---------
;           |OPERADORES
;           |----------
;           |
;   RESIDUO |
```

```
.DEF RESULTADO     = R21
.DEF OPERANDO      = R22
.DEF RESIDUO       = R23
.DEF RADICANDO     = R24
.DEF TEMP          = R25
.DEF TEMP2         = R26
.DEF TEMP3         = R27
.DEF TEMP4         = R3
.DEF TEMP5         = R4
.DEF TEMP6         = R5
.DEF TEMP7         = R6
.DEF RESIDUO_LOW   = R30
.DEF RESIDUO_HIGH  = R31
```

Inicializar LCD, y activar línea 1

```
LDI RADICANDO,255          ;AQUÍ CARGAMOS EL RADICANDO

RCALL CONVIERTE_DATO_A_RADICANDO
RCALL RESOLVIENDO_ENTEROS
RCALL RESOLVIENDO_PUNTO_DECIMAL
RCALL AL_LCD

FIN: RJMP FIN

RESOLVIENDO_ENTEROS:
CLC CLZ CLN
                           ;1a  2a  3a  <--CIFRAS DEL RADICANDO
                           ;R18 R19 R20
LDI TEMP,0                 ;Paso 1: El cuadrado de la primera
                           ;cifra del ;RESULTADO
MOV RESULTADO,TEMP         ;1er RESULTADO PARCIAL
MOV TEMP2,R18

CICLO_PARA_PRIMERA_CIFRA:
MUL TEMP,TEMP
MOV TEMP,R0                ;Copia el resultado del cuadrado a
                           ;"TEMP"
SUB TEMP2,TEMP             ;Resta

BRCS DATO_VALIDO
INC RESULTADO              ;Incrementa RESULTADO hasta dar el
                           ;máximo valor ;necesitado
MOV TEMP, RESULTADO
MOV RESIDUO,TEMP2          ;Copia a RESIDUO la última operación
                           ;antes de CARRY
```

Éstas son las llamadas a subrutina que usaremos en el cálculo para "Pitágoras" más adelante

```
MOV TEMP2,R18
RJMP CICLO_PARA_PRIMERA_CIFRA

DATO_VALIDO:
DEC RESULTADO           ;RESULTADO válido antes del CARRY

;PROCESA EL PRIMER GRUPO DE LA IZQUIERDA DEL RADICAL:

SUMA_PRIMER_GRUPO:      ;Para fusionar el segundo y tercer
                        ;DÍGITO
LDI TEMP,10             ;en la segunda cifra de la derecha
                        ;del RADICANDO
MUL R19,TEMP            ;Posición de DECENAS y UNIDADES
ADD R20,R0              ;Por ello multiplicamos por 10

LDI TEMP,100
MUL RESIDUO,TEMP
ADD R20,R0              ;Sumamos la diferencia de la PRIMERA
                        ;CIFRA
MOV RESIDUO,R20         ;Se actualiza RESIDUO
RJMP ENCONTRANDO_OPERANDOS

ENCONTRANDO_OPERANDOS:
LDI TEMP,2
MOV TEMP2,RESULTADO
MUL TEMP2,TEMP          ;Multiplicamos por 2 el RESULTADO
                        ;para el 1er ;OPERANDO
ADD OPERANDO,R0         ;Copiamos el RESULTADO de la
                        ;multiplicación a ;OPERANDO

;BUSCAMOS LA SEGUNDA CIFRA DEL OPERANDO ("OPER")
;POR MULTIPLICACIONES PROGRESIVAS:

; OPER_0       OPER_1      OPER_2   ...ETC
;x     0      x     1     x     2
; ------       ------      ------


LDI R29,10              ;Valores iniciales para la
                        ;multiplicación sucesiva
LDI R30,0

MOV TEMP3,OPERANDO

MUL TEMP3,R29           ;Multiplicamos por 10 la primera
                        ;cifra del OPERANDO
```

```
MOV TEMP,R0              ;Y lo sumamos al doble de la primera
                         ;cifra de RESULTADO

MULTIPLICACIONES_SUCESIVAS:

MUL R30,TEMP             ;Ingresamos el primer valor de la
                         ;multiplicación
MOV TEMP2,R0             ;Progresiva, donde "S"= 0,1,2,3,...n

MOV TEMP,RESIDUO
SUB TEMP, TEMP2          ;Se resta el RESIDUO menos la
                         ;multiplicación ;anterior

BRCS DATO_VALIDO_2

MOV R28,TEMP             ;Ultima resta viable antes del salto

INC R30

MOV TEMP3,OPERANDO

MUL TEMP3,R29            ;Multiplicamos por 10 la primera
                         ;cifra del OPERANDO
MOV TEMP,R0
ADD TEMP,R30
RJMP MULTIPLICACIONES_SUCESIVAS

DATO_VALIDO_2:
DEC R30
MOV TEMP2,R30
LDI TEMP,10
MUL OPERANDO,TEMP
ADD TEMP2,R0
MOV OPERANDO,TEMP2       ;Se obtiene la segunda cifra del
                         ;OPERANDO
MOV RESIDUO,R28

MUL RESULTADO,TEMP
ADD R30,R0
MOV RESULTADO,R30

CPI R18,0
BREQ ADAPTANDO_OPERANDO
RJMP CONTINUA
```

```
ADAPTANDO_OPERANDO:
LDI TEMP,2
MUL OPERANDO,TEMP
MOV OPERANDO,R0
CONTINUA:
RET

RESOLVIENDO_PUNTO_DECIMAL:
;Hasta aquí se obtuvieron los enteros, el OPERANDO y el
;RESIDUO vamos a borrar el contenido de los registros
;temporales y determinar la parte fraccionaria

CLR TEMP CLR TEMP2 CLR TEMP3    <-- Sintaxis horizontal

;Vamos a agregar DOS-CEROS al RESIDUO
;multiplicando por 100
;Este procedimiento es para 16-Bits

LDI R29,0
LDI R30,10

CICLO_2:
LDI R16,100
MUL RESIDUO,R16
MOV TEMP,R0
MOV TEMP2,R1

;Escribimos el PUNTO DECIMAL en LCD

;Volvemos a hacer las MULTIPLICACIONES SUCESIVAS
;para las cifras después del punto decimal

MUL OPERANDO,R30

ADD R0,R29              ;Debe Ser una Suma de 16-Bits
ADC R1,R2               ;para sumar el CARRY
MOV TEMP3,R1

MUL R0,R29              ;Debe ser una MULTIPLICACIÓN de
                        ;16-Bits
MOV TEMP4,R0
MOV TEMP5,R1

MUL TEMP3,R29           ;TEMP3 es el resultado del anterior
                        ;"R1"
```

```
MOV TEMP6,R0
MOV TEMP7,R1

;RESOLVIENDO LAS MULTIPLICACIONES ANTERIORES:
ADD TEMP6,TEMP5
ADC TEMP7,R2            ;Para sumar el CARRY

;RESULTADO DE MULTIPLICACIONES EN  TEMP7  TEMP6  TEMP4

;RESTANDO
SUB TEMP,TEMP4
SBC TEMP2,TEMP6
;SBC TEMP6,R2           ;Para restar CARRY si existiese

BRCS SIGUE
MOV R19,TEMP            ;R19 valor de RESIDUO
MOV R20,TEMP2           ;R20 valor de RESIDUO

INC R29
CPI R29,10

BREQ SIGUE              ;El límite es 9

RJMP CICLO_2

SIGUE:
.DEF FRACCIONES = R24 <-------- ;VA A MARCAR WARNING PERO
                                ;FUNCIONA BIEN

DEC R29
MOV FRACCIONES,R29
MOV RESIDUO_LOW,R19     ;RESIDUO es de 16-Bits
MOV RESIDUO_HIGH,R20

;EXTRAYENDO LA CIFRA QUE NOS INTERESA
LDI R18,0               ;R18 CENTENAS de RESULTADO
RET

;**************************
;DESPLIEGA LA INFORMACIÓN EN LCD
AL_LCD:
CLR R19 CLR R20 <--------

;EXTRAE LOS DÍGITOS DE RESULTADO:
SALTO_3:
```

```
MOV R0,RESULTADO
SUBI RESULTADO,10

BRCS UNIDADES_RESULTADO
INC R19                  ;R19 DECENAS de RESULTADO

BRCC SALTO_3

UNIDADES_RESULTADO:
MOV RESULTADO,R0

SALTO_4:

SUBI RESULTADO,1

BRCS SUMANDO_PARA_ASCII
INC R20                  ;R20 UNIDADES de RESULTADO

BRCC SALTO_4

;SUMANDO $30 A CADA DÍGITO
SUMANDO_PARA_ASCII:
LDI R16,$30              ;SUMANOS $30
ADD R19,R16              ;ENTERO 1
ADD R20,R16              ;ENTERO 2
ADD FRACCIONES,R16       ;PARTE FRACCIONAL

Sacamos al PORTC:
ENTERO 0
ENTERO 1
ENTERO 2
PUNTO
FRACCIONES

RCALL ESPACIO            ;ESPACIO

RCALL R                  ;R
RCALL E                  ;E
RCALL S                  ;S
RCALL I                  ;I
RCALL D                  ;D

RCALL IGUAL              ;=

;************************************************
```

```
AJUSTA_RESIDUO:
;AJUSTANDO RESIDUO CON FRACCIONES

SUBI FRACCIONES,$30
CPI FRACCIONES,0
BREQ AJUSTE

;EXTRAEMOS CONTENIDO DE RESIDUO PARA 16-Bits
;SI FRACCIONES ES DIFERENTE DE CERO
CLR R19 CLR R20 CLR R21 CLR R22 CLR R23

MOV R0,RESIDUO_LOW
MOV R1,RESIDUO_HIGH

SALTO_5:
SUBI RESIDUO_LOW,100
SBC  RESIDUO_HIGH,R2     ;Por posible CARRY

BRCS DECENAS_RESIDUO
INC R21                  ;R21 CENTENAS de RESIDUO

MOV R0,RESIDUO_LOW
MOV R1,RESIDUO_HIGH

BRCC SALTO_5

DECENAS_RESIDUO:
SALTO_6:

MOV RESIDUO_LOW,R0
MOV RESIDUO_HIGH,R1

SUBI RESIDUO_LOW,10
SBC  RESIDUO_HIGH,R2     ;Por posible CARRY

BRCS UNIDADES_RESIDUO
INC R22                  ;R22 DECENAS de RESIDUO

MOV R0,RESIDUO_LOW
MOV R1,RESIDUO_HIGH

BRCC SALTO_6
UNIDADES_RESIDUO:
MOV RESIDUO_LOW,R0
MOV RESIDUO_HIGH,R1
```

```
SALTO_7:
SUBI RESIDUO_LOW,1
SBC  RESIDUO_HIGH,R2     ;Por posible CARRY

BRCS SUMANDO_PARA_ASCII_2
INC R23                  ;R23 UNIDADES de RESIDUO

BRCC SALTO_7

;******************
;CONDICIONAL: SI FRACCIONES=0
                         ;SIGNIFICA QUE EL RESIDUO NO ES *100
AJUSTE:
MOV R19,RESIDUO_LOW

SALTO_10:
SUBI R19,100
BRCS COMPENSANDO
INC R17                  ;R17 RESIDUO ÚNICO
BRCC SALTO_10

COMPENSANDO:
MOV RESIDUO_LOW,R17
LDI RESIDUO_HIGH,0

LDI R16,$30
ADD RESIDUO_LOW,R16

OUT PORTC, RESIDUO_LOW

LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
;************************************************

SUMANDO_PARA_ASCII_2:
LDI R16,$30
ADD R21,R16
ADD R22,R16
ADD R23,R16
```

```
Sacamos al PORTC:
R21 CENTENAS de RESIDUO
R22 DECENAS de RESIDUO
R23 UNIDADES de RESIDUO

RET
```

```
;**************************
;EXTRAYENDO DÍGITOS
CONVIERTE_DATO_A_RADICANDO:

LDI R18,0               ;CENTENAS
LDI R19,0               ;DECENAS
LDI R20,0               ;UNIDADES

LEYENDO_HUNDREDS:
MOV R0,RADICANDO
SUBI RADICANDO,100

BRCS LEYENDO_DECIMALES
INC R18                 ;CENTENAS

BRCC LEYENDO_HUNDREDS

LEYENDO_DECIMALES:
MOV RADICANDO,R0

SALTO:
MOV R0,RADICANDO
SUBI RADICANDO,10

BRCS LEYENDO_UNIDADES
INC R19                 ;DECENAS

BRCC SALTO

LEYENDO_UNIDADES:
MOV RADICANDO,R0

SALTO_2:
MOV R0,RADICANDO
SUBI RADICANDO,1
```

```
BRCS RESOLVIENDO_ENTEROS_PARCHE
INC R20                    ;UNIDADES

BRCC SALTO_2

RESOLVIENDO_ENTEROS_PARCHE:
RET

.INCLUDE "CODIGOS_ASCII_Y_CONTROL_LCD.TXT"
.INCLUDE  "DELAY.TXT"
```

Programa:

```
;PROGRAMA PARA CALCULAR LA HIPOTENUSA DE UN TRIÁNGULO RECTÁN-
GULO
;USANDO LA ECUACIÓN DE PITÁGORAS
```

Encabezado para ATmega8515

Stack Pointer para ATmega8515

Inicializar LCD, y activar línea 1

```
;DEBE HACER EL MAPEO DE REGISTROS PARA RESOLVER ESTE PROGRAMA.
;AQUÍ USamos R16 Y R17 POR CONVENIENCIA.
;HAGAMOS b=10 Y c=10

IN R16,PINA                ;Para variable "b"
IN R17,PINB                ;Para variable "c"

MUL R16,R16                ;Multiplicamos b*b
MOV R18,R0
```

```
MUL R17,R17             ;Multiplicamos c*c
MOV R19,R0

ADD R18,R19             ;El RESULTADO es 200 (8-bits)
MOV R24,R18             ;Recuerde que usamos R24 para el
                        ;RADICANDO
.
.
.DEF RADICANDO =R24     ;LDI RADICANDO,200
.
.
RCALL CONVIERTE_DATO_A_RADICANDO
RCALL RESOLVIENDO_ENTEROS
RCALL RESOLVIENDO_PUNTO_DECIMAL
RCALL AL_LCD
FIN: RJMP FIN
```

El resultado en el display será:

```
ENTERO_1     =1
ENTERO_2     =4
FRACCIONES =1
RESIDUO      =159
```