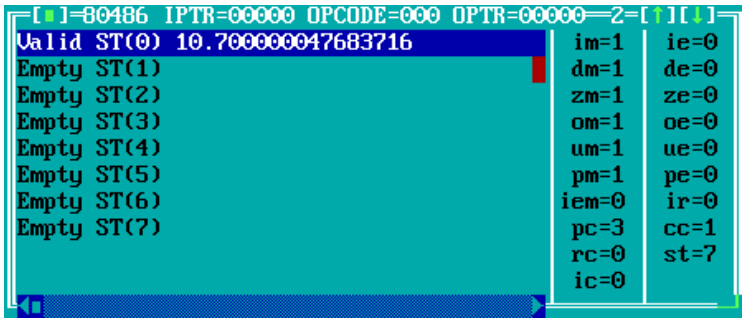
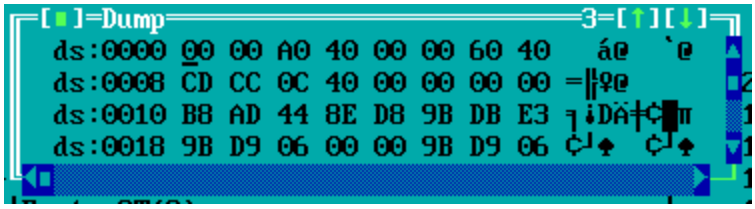
Lab 2

1.Write an ALP to do addition of three floating point numbers using 8087 instruction set.
Test case data: x = 3.5 y = 5.0 z = 2.2

```
data segment
    a dd 5.0
    b dd 3.5
    c dd 2.2
    d dd ?

data ends

code segment
    assume cs:code, ds:data
    start:
    mov ax,data
    mov ds,ax

    finit
    fld a
    fld b
    fadd
    fld c
    fadd
    fst d
    int 3

    code ends
end start
```

Output:



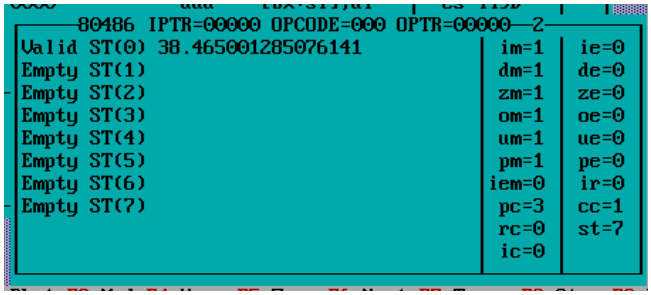


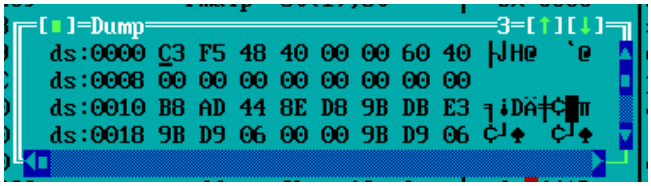
2.Write an ALP to find area of a circle using 8087 instruction set.
Test case data: Pi = 3.1472 radius = 3.5

sample.asm sec.asm

```
1   data segment
2       pi dd 3.14
3       r dd 3.5
4       res dd ?
5   data ends
6   code segment
7       assume cs:code,ds:data
8       start:
9       mov ax,data
10      mov ds,ax
11      finit
12      fld pi
13      fld r
14      fld r
15      fmul
16      fmul
17      fst res
18      int 3
19  code ends
20  end start
```

Output:

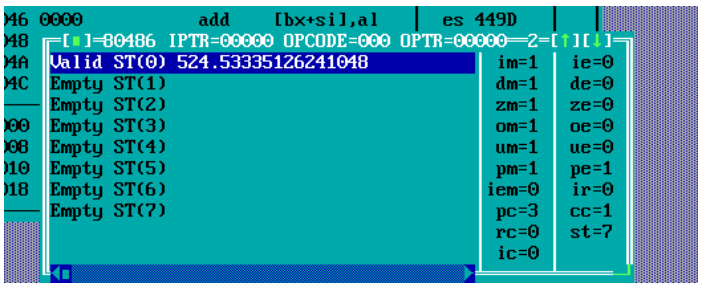


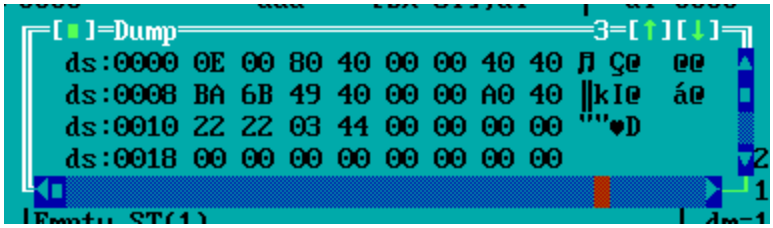


3. Write an ALP to find volume of sphere using 8087 instruction set.
Test case data: Pi = 3.1472 radius = 5.0

```
data segment
    a dd 4.0
    b dd 3.0
    pi dd 3.1472
    r dd 5.0
    vol dd ?

data ends

code segment
    assume cs:code, ds:data
    start:
    mov ax,data
    mov ds,ax

    finit
    fld a
    fld b
    fdiv
    fld pi
    fmul
    fmul r
    fmul r
    fmul r
    fst vol
    int 3

    code ends
end start
```

Output:

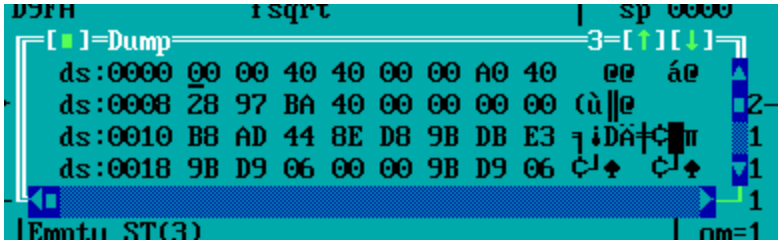




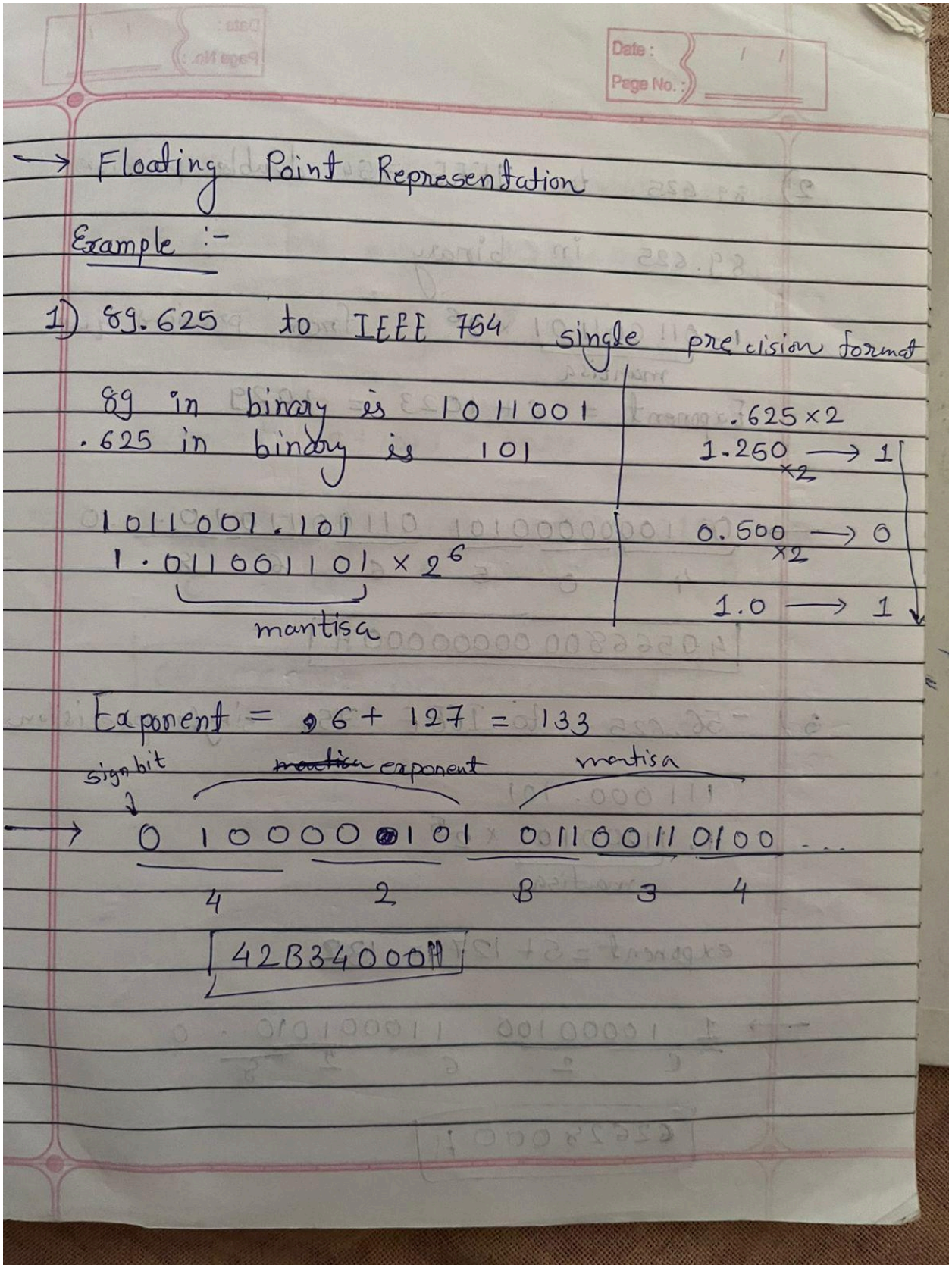
4.Write an ALP to find c = sqrt(a^2 + b^2)
Test case data: a = 5.0 b = 3.0

```
data segment
    a dd 3.0
    b dd 5.0
    res dd ?
data ends
code segment
    assume cs:code,ds:data
    start:
    mov ax,data
    mov ds,ax
    finit
    fld a
    fld a
    fmul
    fld b
    fld b
    fmul
    fadd
    fsqrt
    fst res
    int 3
code ends
end start
```

Output:


fsqrt
sp 0000
[■]=Dump
3=[↑][↓]
ds:0000 00 00 40 40 00 00 A0 40 @@ á@
ds:0008 28 97 BA 40 00 00 00 00 (ù╠@
ds:0010 B8 AD 44 8E D8 9B DB E3 ╕¡DÄ╪¢█π
ds:0018 9B D9 06 00 00 9B D9 06 ¢┘♠ ¢┘♠
Empty ST(3)
cm=1

Floating Point Representation:



→ Floating Point Representation

Example :-

1) 89.625 to IEEE 754 single precision format

89 in binary is 1011001

.625 in binary is 101

.625 × 2
1.250 → 1 (×2)
0.500 → 0 (×2)
1.0 → 1

1011001.101

1.011001101×2^6

mantisa

Exponent = 6 + 127 = 133

sign bit | exponent | mantisa

→ 0 | 10000101 | 01100110100...

4 2 B 3 4

42B34000H

Date :
Page No. :

2) 89.625 to IEEE 754 double precision

89.625 in binary

1.011001101×2^6 from previous que

mantisa

Exponent = 6 + 1023 = 1029

→ 0 10000000101 01100110100...0

4 0 5 6 6 8

40566800 00000000 H

3) -56.625 to IEEE 754 single precision

111000.101

1.11000101×2^5

mantisa

exponent = 5 + 127 = 132

→ 1 10000100 11000101000...0

C 2 6 2 8

C2628000 H

Date :
Page No. :

4) -56.625 to IEEE 754 double precision

→ -56.625 in binary 111000.101

$1.\underbrace{11000101}_{\text{mantisa}} \times 2^5$

exponent = 5 + 1023 = 1028

→ 1 10000000100 11000101000...0

C 0 4 C 5 0

C04C5000 00000000 H