**PART 3**

**ALGORITHM**

GENERAL STEPS:

You should look the first and the second number.

If first number is bigger than second number you should change their locations.

However, if the first number is smaller than the second number, you don’t change their locations.

You should repeat these directions such that you should control the first and the second than second and third.

You should continue like this.

You can turn the first number if finish numbers and it isn’t true.

Finally you continue doing same things.

FOR EXAMPLE:

I choose 1 5 7 3 4 9

I look the first and the second 1<5 so it is true I continue controlling the numbers

1 5 7 3 4 9

Then I continue second and third

1 5 7 3 4 9

It is true

1 5 7 3 4 9

It is not true so you should change their location

1 5 3 7 4 9

7>4 so you should change

1 5 3 4 7 9

7<9 it is true so you can turn false number

1 5 3 4 7 9

1 3 5 4 7 9

5>3 so you change their locations

5>4 so you again change five and four

1 3 4 5 7 9

Finally I find the correct sorting

1 3 4 5 7 9

**PART 2**

**IEEE 32 BITS FLOATING POINT FORMAT**

**1)**

|  |  |  |
| --- | --- | --- |
| **SIGN** | **EXPONENT** | **MANTISSA** |
| 1 Bit | 8 Bit | 23 Bit |

The number is pozitif so sign is 0.

If the number is negative, it have to be 1.

Exponent should be 8 bit.

We can find 98\*10⁻⁴

We can find exponent such that we use (-) and 4 of 10⁻⁴

Mean of (-) “we should start 1” than we should create 4 by binary system.

It become 100 than our exponent is ready:

10000100

Mantissa consist 23 bit.

We will use 98 such that we have to write 98 by binary system.

If we write 98 by binary system, we can find

Thus, we find it:

|  |  |  |
| --- | --- | --- |
| **Sign** | **Exponent** | **Mantissa** |
| 0 | 10000100 | 00000000000000001100010 |

2)

1 10000010 11100000000000000000000

The first number indicate positive or negative so our number is 1 if it is negative

For 1000010 =130

For 11100000000000000000000 =½ +¼ +⅛

= 0.875

Our formula is

(-1)\*(1+0.875)\*2 ¹³°⁻¹²⁷

=-1\*1.875\*8

= -5.625

We can find the answer -5.625