\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw3\_1

Author: F74055047 張財實 <[changchaishi@gmail.com](mailto:changchaishi@gmail.com)>

Class: 乙班

Description:

Learned to use testing a bit to print 0 or 1 from chapter 20.We have to use pointer to treat the float number as an integer thus the bitwise operation can take place.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

#define bias 127

int main(int argc,char \*argv[]){

int N;//select int for cloat

N=atoi(argv[1]);

//int to bit pattern

if(N==1){

int X,i;

X=atoi(argv[2]);

for(i=31;i>=0;i--)

//shift to the 1st bit from left to right

if(X&1<<i)

//testing a bit

printf("1");

//print bit

else

printf("0");

printf("\n");

}

else if(N==2){

int i,x,\*A;

float X;

x=atoi(argv[2]);

X=x;

//argument input is a integer

A=(int\*)&X;

//make float point acts as integer

for(i=31;i>=0;i--)

//shift to the 1st bit from left to right

if(\*A&1<<i){

//testing a bit

printf("1");

}

//print bit

else{

printf("0");

}

printf("\n");

}

return 0;

}

Compilation:

gcc -o hw3\_1 hw3\_1.c

Execution:

./hw3\_1 2 100

Output:

01000010110010000000000000000000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw3\_2

Author: F74055047 張財實 <[changchaishi@gmail.com](mailto:changchaishi@gmail.com)>

Class: 乙班

Description:

This question is all about reverse the hw3\_1 b) question. First I have to use 2-D array to assign 32 bit. By the time of assignment, use ‘if’ to check which array has character 1. Set initial float value be 0.0, and use int pointer to ‘setting a bit’, the float value is being updated by the loop of shifting bits. Finally just print the decimal value out.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

#define bias 127.0f //bias is 127

int main(int argc,char \*argv[]){

int i,\*A;

float value;

value=0.0f;

//value of bit pattern initiate with 0,with 32 0s

A=(int\*)&value;

//use pointer A to act float as int

for(i=0;i<31;i++){

if((argv[1][i]-'0')==1)

//same as atoi

\*A|=1<<(31-i);

//value getting set everytime it meets the condition

//when ith array is 1,to set the bit,we shift by (31-i)

}

printf("%.23e\n",value);//print the value

}

Compilation:

gcc -o hw3\_2 hw3\_2.c

Execution:

./hw3\_2 00000000100000000000000000000000

Output:

1.17549435082228750796874e-38

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw3\_3

Author: F74055047 張財實 <[changchaishi@gmail.com](mailto:changchaishi@gmail.com)>

Class: 乙班

Description:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Smallest value: 1.1754944E-38

Bit pattern: 00000000100000000000000000000000

Largest value: 3.4028235E38

Bit pattern: 01111111011111111111111111111111

Special Exponent Values with All 0’s and All 1’s

|  |  |  |  |
| --- | --- | --- | --- |
| Sign bit | Exponent | Mantissa | Meaning |
| 0 | All 0’s | All 0’s | +0 |
| 1 | All 0’s | All 0’s | -1 |
| 0 | All 0’s | Not 0 | +Denormal |
| 1 | All 0’s | Not 0 | -Denormal |
| 0 | All 1’s | All 0’s | +Infinity |
| 1 | All 1’s | All 0’s | -Infinity |
| 0 | All 1’s | Not 0 | NaN |
| 1 | All 1’s | Not 0 | NaN |

NaN- 01111111101010000000000000000000

Inf- 01111111100000000000000000000000

Referrence:

https://www.youtube.com/watch?v=H79PNQ4Z9HE

https://www.h-schmidt.net/FloatConverter/IEEE754.html