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

Report: HW4

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

Description:

How do you finish this homework? Stackoverflow

What did you learned from this homework? 這次查的東西太多了，已經不知道該如何列舉。花最多時間的應該在研究每個資料型別要怎麼對齊，還有一些數字莫名其妙就overflow。

Did you do or write something special for bonus? 沒☹

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

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

int float\_to\_bit (int a, int location)

{

int bit = a & 1<<location;

if(bit == 0) return 0;

else return 1;

}

int double\_to\_bit (long long a, long long location)

{

long long b = 1;

long long bit = a & b<<location;

if(bit == 0) return 0;

else return 1;

}

int intBinary\_to\_decimal (long long n)

{

int decimal = 0, i = 0, remain;

while(n!=0)

{

remain = n%10;

n /= 10;

decimal += remain\*pow(2,i);

++i;

}

return decimal;

}

double floatBinary\_to\_decimal (double n)

{

double decimal = 0;

int i;

for(i=-1; i>=-23; i--)

{

n \*= 10;

if(n > 1){

n -= 1;

decimal += pow(2,i);

}

}

return decimal;

}

double doubleBinary\_to\_decimal (double n)

{

double decimal = 0;

int i;

for(i=-1; i>=-52; i--)

{

n \*= 10;

if(n > 1){

n -= 1;

decimal += pow(2,i);

}

}

return decimal;

}

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

{

int s = (int)atoi(argv[1]);

if(s == 1){ //float number to bit pattern

float a = (float)atof(argv[2]);

int \*b;

b = &a;

int i;

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

printf("%d",float\_to\_bit(\*b,i));

}

else if(s == 2){ //double number to bit pattern

double a = (double)atof(argv[2]);

long long \*b;

b = &a;

int i;

for(i=63; i>=0; i--)

printf("%d",double\_to\_bit(\*b,i));

}

else if(s == 3){ //bit pattern (float) to float number

int sig = (int)atoi(argv[2]);

long long exp = (long long)atoll(argv[3]);

double fra = (double)atof(argv[4]) \* pow(10,-23);

int e = intBinary\_to\_decimal(exp) - 127;

double f = floatBinary\_to\_decimal(fra) +1;

double floatNumber = f \* pow(2,e) \* pow(-1,sig);

printf("%lf\n",floatNumber);

}

else if(s == 4){ //bit pattern (double) to float number

int sig = (int)atoi(argv[2]);

long long exp = (long long)atoll(argv[3]);

double fra = (double)atof(argv[4]) \* pow(10,-52);

int e = intBinary\_to\_decimal(exp) - 1023;

double f1 = doubleBinary\_to\_decimal(fra);

double f = f1 +1;

double floatNumber = f \* pow(2,e) \* pow(-1,sig);

printf("%lf\n",floatNumber);

}

else{

printf("you put wrong s");

}

return 0;

}

Compilation:

gcc -g hw4.c -o hw4 -lm

Execution:

./hw4 1 85.125

./hw4 2 85.125

./hw4 3 0 10000101 01010100100000000000000

./hw4 4 0 10000000101 0101010010000000000000000000000000000000000000000000

Output:

01000010101010100100000000000000

0100000001010101010010000000000000000000000000000000000000000000

85.125237

85.125000

2

2-1 (yes/no), If not, what is the smallest floating point number?

No. The smallest positive number is 2-149

2-2 (bit pattern)

0 00000000 00000000000000000000000

2-3 explanation

我猜是overflow吧。後面的數字沒有被存進double裡，所以只剩1.1754943508222875。

2-4 (bit pattern,選其中一個即可)

0 11111111 00000000000000000000000 infinity

2-5 i.number

3.141592

ii.number

3.333333