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

Report: HW7

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

Class: 物理系

Description:

How do you finish this homework? 老師的PPT

What did you learned from this homework? Struct和union的用法

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

int intBinary\_to\_decimal (long long n);

double floatBinary\_to\_decimal (double n);

double doubleBinary\_to\_decimal (double n);

typedef union{

float f;

double d;

struct{

double mantissa;

long long exponent;

int sign;

} bitpattern;

} u\_float;

struct s\_float{

float f;

double d;

double mantissa;

long long exponent;

int sign;

};

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

{

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

u\_float uf;

struct s\_float sf;

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

uf.f = (float)atof(argv[2]); //union

int \*b;

b = &uf.f;

int i;

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

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

printf("\n");

sf.f = (float)atof(argv[2]); //struct

int \*a;

a = &sf.f;

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

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

printf("\n");

}

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

uf.d = (double)atof(argv[2]); //union

long long \*b;

b = &uf.d;

int i;

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

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

printf("\n");

sf.d = (double)atof(argv[2]); //struct

long long \*a;

a = &sf.d;

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

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

printf("\n");

}

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

uf.bitpattern.sign = (int)atoi(argv[2]); //union

uf.bitpattern.exponent = (long long)atoll(argv[3]);

uf.bitpattern.mantissa = (double)atof(argv[4]) \* pow(10,-23);

int e = intBinary\_to\_decimal(uf.bitpattern.exponent) - 127;

double f = floatBinary\_to\_decimal(uf.bitpattern.mantissa) +1;

uf.d = f \* pow(2,e) \* pow(-1,uf.bitpattern.sign);

printf("%.3lf\n",uf.d);

sf.sign = (int)atoi(argv[2]); //struct

sf.exponent = (long long)atoll(argv[3]);

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

e = intBinary\_to\_decimal(sf.exponent) - 127;

f = floatBinary\_to\_decimal(sf.mantissa) +1;

sf.d = f \* pow(2,e) \* pow(-1,sf.sign);

printf("%.3lf\n",uf.d);

}

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

uf.bitpattern.sign = (int)atoi(argv[2]); //union

uf.bitpattern.exponent = (long long)atoll(argv[3]);

uf.bitpattern.mantissa = (double)atof(argv[4]) \* pow(10,-52);

int e = intBinary\_to\_decimal(uf.bitpattern.exponent) - 1023;

double f1 = doubleBinary\_to\_decimal(uf.bitpattern.mantissa);

double f = f1 +1;

uf.d = f \* pow(2,e) \* pow(-1,uf.bitpattern.sign);

printf("%lf\n",uf.d);

sf.sign = (int)atoi(argv[2]); //struct

sf.exponent = (long long)atoll(argv[3]);

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

e = intBinary\_to\_decimal(sf.exponent) - 1023;

f1 = doubleBinary\_to\_decimal(sf.mantissa);

f = f1 +1;

sf.d = f \* pow(2,e) \* pow(-1,sf.sign);

printf("%lf\n",sf.d);

}

else{

printf("you put wrong s");

}

return 0;

}

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;

}

Compilation:

gcc -g hw7.c -o hw7 -lm

Execution:

./hw7 1 85.125

./hw7 2 85.125

./hw7 3 0 10000101 01010100100000000000000

./hw7 4 0 10000000101 0101010010000000000000000000000000000000000000000000

Output:

01000010101010100100000000000000

01000010101010100100000000000000

0100000001010101010010000000000000000000000000000000000000000000

0100000001010101010010000000000000000000000000000000000000000000

85.125

85.125

85.125

85.125