# 00P 第十四周作业文档

2019010175 孔瑞阳 计科 91

## 一、项目信息

## 1、功能说明

输入一个双精度浮点数,输出它在内存中的二进制表示。 输出非数和无穷的二进制表示,并计算它们之间的四则运算。

#### 2、软件构件介绍

文件	功能介绍
CP_UnionDoubleUnsignedlonglong.h/cpp	double 的共用体类
CP_UnionDoubleUnsignedlonglongMain.cpp	主程序

#### 3、测试环境

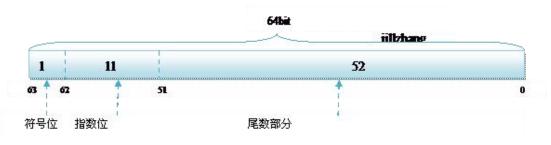
СРИ	Intel(R) Core(TM)i7-9750H CPU @ 2.6Ghz 6 核 12 线程			
GPU	NVIDIA GeForce RTX2070			
RAM	DDR4 16G+16G			
Operating System Microsoft Windows 版本 1909				
Compiler	MSVC++ 14.24			

## 二、程序过程

- 1、将输出的 double 在内存中的值转换成 unsigned long long。
- 2、通过 bitset 的构造函数,将 unsigned long long 转化成 bitset。
- 3、通过 bitset 的 to\_string 成员函数,将 bitset 转化为 string。
- 4、输出最后的 string。

# 三、测试

## 1、double的二进制表示



Source: https://blog.csdn.net/yansmile1/article/details/70145416/

一个浮点数可以唯一表示为: (+/-)(1+x)\*2<sup>y</sup> (0<=x<1) 符号位 0 表示+, 1 表示-。

指数位表示 y+127 的值。

尾数位表示x。

例如: 8.5 = +(1+0.0625)\*2<sup>3</sup> = +(1+(0b)0.0001)\*2<sup>3</sup>

所以 x 的前缀为 0001

符号位为 3+127=130=(0b)1000000010

所以8.5的 double 二进制表示为

## 2、测试

数的	选取	丰二	输出			
类型	案例	表示				
零	0	/	0b <b>0</b> 00000000000			
<b>*</b>	U	/	000000000000000000000000000000000000000			
名示	0	/	0b 10000000000			
贝令	负零		000000000000000000000000000000000000000			
	1024	<del>+</del> (1+ <b>0</b> )	0b <b>0</b> 10000001001			
正数	1024 *2 <sup>10</sup>		000000000000000000000000000000000000000			
上级	4/3 +(1+ <b>1/3</b> ) *2 <sup>0</sup>		4/2	<b>+</b> (1+ <b>1</b> / <b>3</b> )	Оь <mark>0</mark> 01111111111	
			01			
	0 5	<del>-</del> (1+0.0625)	0b 11000000010			
<b>名业</b>	-8. 5	*2 <sup>3</sup>	000100000000000000000000000000000000000			
负数	_1 /6	<b>-</b> (1+ <b>1</b> / <b>3</b> )	ОЬ 101111111100			
	-1/6	*2 <sup>-3</sup>	01			

都与上述预测的二进制表示相同。

## 四、提高部分

## 1、非数和无穷的表示

测试结果如下:

数据类型	二进制表示
inf	0ь 011111111111100000000000000000000000
-inf	0ь 111111111111100000000000000000000000
-nan(ind)	0ь 111111111111110000000000000000000000
nan	0b <b>011111111111110000000000000000000000</b>

符号位依然表示正负。

指数位全是1。

尾数位若全是 0, 表示 inf。若不全为 0, 表示 nan。

## 2、非数和无穷的四则运算

+	inf	-inf	-nan (ind)	nan	1
inf	inf	-nan(ind)	-nan (ind)	nan	inf
-inf	-nan(ind)	-inf	-nan (ind)	nan	-inf
−nan (ind)	-nan(ind)	-nan (ind)	-nan (ind)	-nan (ind)	-nan (ind)
nan	nan	nan	nan	nan	nan
1	inf	-inf	-nan(ind)	nan	2

-	inf	-inf	−nan (ind)	nan	1
inf	-nan(ind)	inf	-nan(ind)	nan	inf
-inf	-inf	-nan (ind)	-nan(ind)	nan	-inf
-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)
nan	nan	nan	nan	nan	nan
1	-inf	inf	-nan(ind)	nan	0

*	inf	-inf	-nan(ind)	nan	1
inf	inf	-inf	-nan(ind)	nan	inf
-inf	-inf	inf	-nan (ind)	nan	-inf
-nan (ind)	-nan(ind)	-nan(ind)	-nan (ind)	-nan (ind)	-nan(ind)
nan	nan	nan	nan	nan	nan
1	inf	-inf	-nan (ind)	nan	1

/	inf	-inf	−nan (ind)	nan	1
inf	-nan(ind)	-nan(ind)	-nan(ind)	nan	inf
-inf	-nan(ind)	-nan (ind)	-nan (ind)	nan	-inf
−nan (ind)	-nan(ind)	-nan(ind)	-nan (ind)	-nan(ind)	-nan(ind)
nan	nan	nan	nan	nan	nan
1	0	-0	-nan (ind)	nan	1

#### 总结:

- 1、nan 与 inf 或者通常数计算,结果都是 nan。
- 2、nan 与-nan(ind)运算, 谁在左边结果是谁。
- 3、inf表示 lim n->inf n, -inf表示 inf表示 lim n->inf -n, nan表示不存在。 则四则运算的结果就是这两个极限的运算结果。