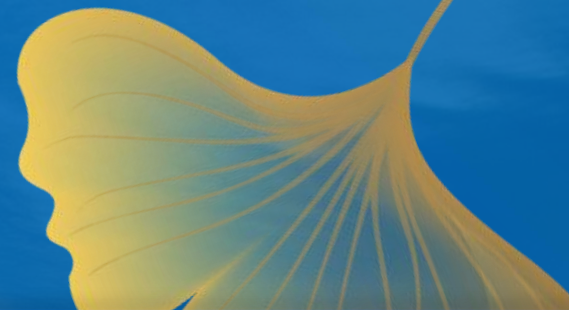




南京大学120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022

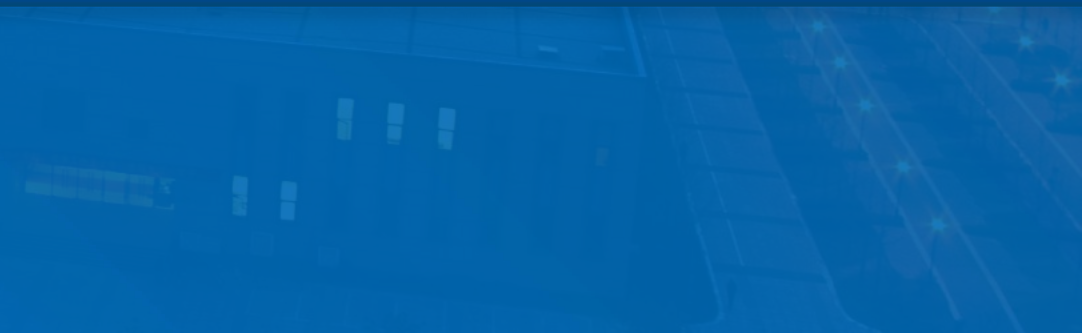
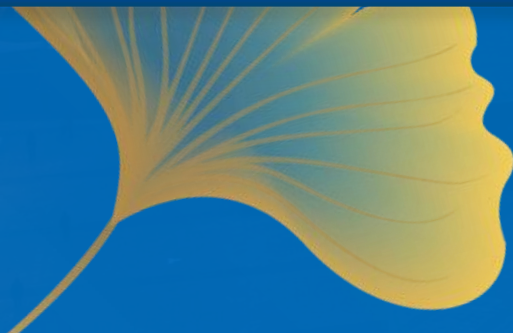
1902 2022



# 数据的机器级表示-2

Data Representation at Machine Level

李杉杉





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022

# CONTENT

## 目录

**01** 浮点数数据类型

**02** ASCII编码

**03** C语言中的数据类型





南京大学120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022

# 01 浮点数数据类型



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 小数——定点小数

- 十进制，小数点，定点表示法 ( *fixed point* )

- $d_k d_{k-1} \dots d_1 d_0 . d_{-1} d_{-2} \dots d_{-m}$

- $3456.78 = 3 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 + 7 \cdot 10^{-1} + 8 \cdot 10^{-2}$

$$d = \sum_{i=-m}^k 10^i \times d_i$$

- 二进制

- $b_k b_{k-1} \dots b_1 b_0 . b_{-1} b_{-2} \dots b_{-m}$

- 00101000.101

$$\begin{aligned} &= 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \\ &\quad + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 32 + 8 + 0.5 + 0.125 = 40.625 \end{aligned}$$

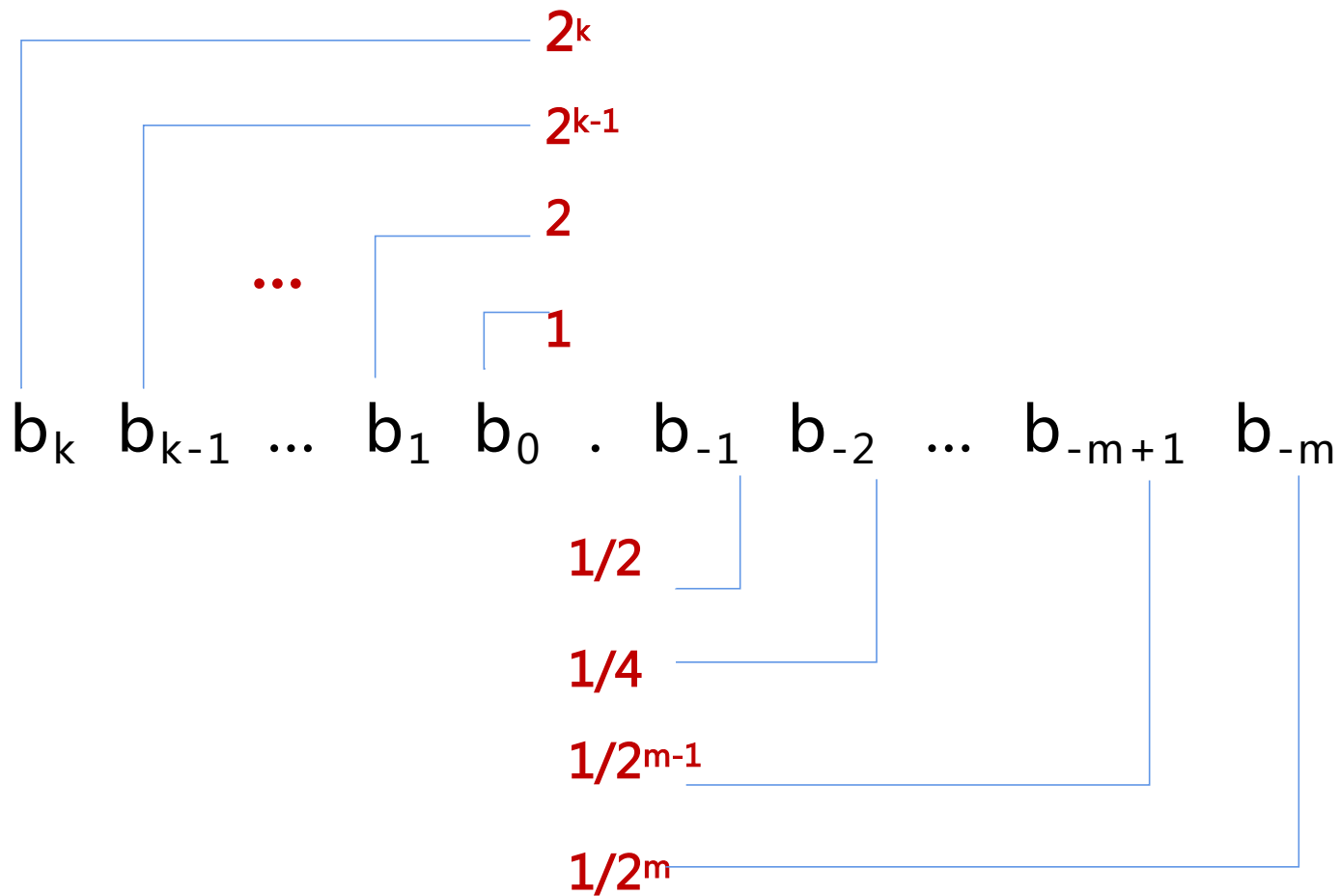
$$b = \sum_{i=-m}^k 2^i \times b_i$$



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 定点小数——精度

- 如果十进制小数不能用有限位的二进制数表示
- 则根据精度取几位
- 示例：
  - $(0.414)_{10} \approx (0.01101)_2$  (取5位)
  - 或写为  $0.414D \approx 0.01101B$  (取5位)



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 定点小数——数值范围

- 如果使用到超出位数表示范围的小数，例如阿佛加德罗常数  $6.023 \times 10^{23}$  ？
- 32位补码整数的数值范围：
  - $-2147483648 \sim 2147483647$ ，即  $-2^{31} \sim +2^{31}-1$
  - 精度31位
- 为了表示  $6.023 \times 10^{23}$ 
  - 数值范围——需要79位（二进制补码整数）
  - 精度——只需要4个十进制数（6023）
- 问题：表示精度的位数过多，而表示范围的位数却不足？



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 小数——浮点小数

- 小数点可以浮动，与科学计数法类似
- 十进制的科学计数法

分/尾数 ( mantissa )      阶/指数 ( exponent )

符号 ( sign )      基/底 ( radix/base )

$$-3.04 \times 10^{20}$$

- 二进制的科学计数法

分/尾数 ( mantissa )      阶/指数 ( exponent )

符号 ( sign )      基/底 ( radix/base )

$$-1.01 \times 2^{20}$$

基底不变，对符号、尾数和阶分别编码即可以表示一个浮点数





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 小数——浮点小数

- 二进制的科学计数法
- 任何浮点数都能以多种样式来表示

$$0.110 \times 2^5, 110 \times 2^2, 0.0110 \times 2^6$$

- 规格化表示

$$\pm 1.bbb\dots b \times 2^E$$

$$V = (-1)^S \times M \times 2^E$$

- 包括
  - 符号 ( S )
  - 分/尾数 ( 精度 , M , 原码 )
  - 基/底数 ( R=2 )
  - 指/阶数 ( 范围 , E , 移码 )



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 拓展：移码

- 将每一个数值加/减一个**偏置常数** ( excess/bias )
- 通常当编码位数为n时，bias取 $2^{n-1}$ 或 $2^{n-1} - 1$  ( 如IEEE 754 )
- 例如n=4时，bias=8
  - 3的移码=3+8=11  $\Rightarrow$  1011
  - -3的移码=-3+8=5  $\Rightarrow$  0101

真值	二进制	真值	二进制
0	1000	-0	1000
1	1001	-1	0111
2	1010	-2	0110
3	1011	-3	0101
4	1100	-4	0100
5	1101	-5	0011
6	1110	-6	0010
7	1111	-7	0001
		-8	0000

当bias为 $2^{n-1}$ ，  
移码和补码仅第一位不同  
0的表示唯一



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 小数——浮点小数

$$V = (-1)^S \times M \times 2^E$$

- $E$  使用**移码**：例如计算  $1.01 \times 2^{-1} + 1.11 \times 2^3$  时，需要将**低阶**（ $2^{-1}$ ）转为**高阶**（ $2^3$ ），那么首先需要比较**-1**和**3**的大小；实现负数阶码效果，而不像补码表示还要考虑符号
- $M$  使用**原码**：对称，无符号时原码的乘法运算简单
- 对于一定长度的规格化数，表示范围和精度之间存在权衡（**总位数不变**）
  - **增加**阶数（ $E$ ）位数：**扩大**表示范围，**降低**表示精度
  - **增加**尾数（ $M$ ）位数：**减少**表示范围，**提高**表示精度
  - **更大的底**（ $B$ ）：如4，8，16，**扩大**表示范围，**降低**表示精度

统一的标准？



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# 浮点数数据类型

## IEEE-754浮点数

- IEEE（国际电气和电子工程师协会）浮点数算术运算标准
  - 二进制表示的浮点数格式标准
  - IEEE 754-1985 [https://en.wikipedia.org/wiki/IEEE\\_754-1985](https://en.wikipedia.org/wiki/IEEE_754-1985)
  - 两种主要的浮点数格式：单精度（32位）和双精度（64位）
- 提出历史
  - 20世纪70年代，IEEE成立委员会着手制定浮点数标准
  - 20世纪80年代，Intel邀请Kahan教授设计8087处理器的浮点运算单元
  - IEEE邀请Kahan基于8087中的浮点标准起草一份通用标准
  - 1985年提出浮点数标准IEEE 754



Larger Photo

**William M. Kahan**

Professor Emeritus

Research Areas

[Computer Architecture & Engineering \(ARC\)](#)

[Scientific Computing \(SCI\)](#)

Computer architecture; Scientific computing;  
Numerical analysis



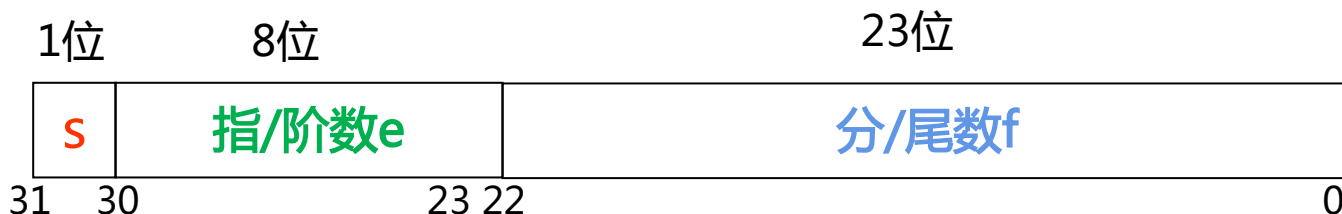
南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754单精度浮点数 ( float )

$$V = (-1)^S \times M \times 2^E$$



- 32 位
- 符号 ( S )
  - 0代表正数，1代表负数
- 指/阶数 ( e ) :
  - 无符号整数，[0, 255] ( 8位 )
- 分/尾数 ( f ) :
  - 第0-22位 ( 23位 )
  - 精度约7位小数 ( 十进制数 )， $2^{23}=8388608$

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{e-127}, & 1 \leq e \leq 254 \\ (-1)^s \times 0.f \times 2^{-126}, & e = 0 \\ (-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), & e = 255, f = 0 \\ \text{NaN(Not a Number)}, & e = 255, f \neq 0 \end{cases}$$



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754单精度浮点数 ( float )

- 规格化 ( M第一位为1 , 阶数不为0 )

S	$\neq 0$ and $\neq 255$	f
---	-------------------------	---

$$V = (-1)^S \times M \times 2^E$$

- $0 (00000000) < e \leq 254 (11111110)$
- $E = e - \text{bias}$
- $\text{bias}(\text{float}) = 2^{8-1}-1 = 127$
- $-126 \leq E \leq 127$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{128} = \text{Infinity} (+ \text{and } -), e = 255, f = 0 \\ \text{NaN} (\text{Not a Number}), e = 255, f \neq 0 \end{cases}$$

f <sub>22</sub>	f <sub>21</sub>	.....	f <sub>1</sub>	f <sub>0</sub>
-----------------	-----------------	-------	----------------	----------------

$$M = 1.f_{22}f_{21}...f_1f_0 = 1 + f$$

精度实际为24位二进制数 ( 隐藏位为1 , 忽略 )



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754单精度浮点数 ( float )

- 非规格化 ( M第一位为0 , 真值等于或接近0 , 阶数为0 )

S	00000000	f
---	----------	---

$$V = (-1)^S \times M \times 2^E$$

- 用于表示0

- $s=0$   $M = 0+f = 0$  ( 隐藏位为0 )
- $\rightarrow X = +0.0$
- $s=1$   $M = 0+f = 0$  ( 隐藏位为0 )
- $\rightarrow X = -0.0$

- 用于表示接近0的数值

- $M = 0 + f \neq 0$  ( 隐藏位为0 )
- $E = 1 - \text{bias}$
- $\text{bias}(\text{float}) = 2^{8-1} - 1 = 127$
- $E = -126$

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{e-127}, & 1 \leq e \leq 254 \\ (-1)^s \times 0.f \times 2^{-126}, & e = 0 \\ (-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), & e = 255, f = 0 \\ \text{NaN}(\text{Not a Number}), & e = 255, f \neq 0 \end{cases}$$



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754单精度浮点数 ( float )

- 无穷大 ( 阶数最大 , M第一位为1 , f为0 )

S	11111111	000000000000000000000000
---	----------	--------------------------

- $s=0, f=0, X=+\infty$

- $s=1, f=0, X=-\infty$

- $E = e - \text{bias} = 255 - 127 = 128$

- $M = 1 + f = 1.0000\dots$

$$V_{\text{float}} = \{$$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity} (+ \text{and } -), e = 255, f = 0$$

$$\text{NaN (Not a Number)}, e = 255, f \neq 0$$

- 非数值 ( 阶数最大 , f不为0 )

S	11111111	$\neq 0$
---	----------	----------

- $0.0/0.0 ; 0.0 \times \infty ; \infty / \infty ; \infty - \infty$

- 仅仅是一种特殊状态标记 , NaN





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754单精度浮点数 ( float )

- 规格化 ( 非零实数 )

S	≠ 0 and ≠ 255	f
---	---------------	---

- 非规格化 ( 非常小的实数或零 )

S	00000000	f
---	----------	---

- 无穷

S	11111111	000000000000000000000000000000
---	----------	--------------------------------

- 非数值

S	11111111	≠ 0
---	----------	-----

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{e-127}, & 1 \leq e \leq 254 \\ (-1)^s \times 0.f \times 2^{-126}, & e = 0 \\ (-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), & e = 255, f = 0 \\ \text{NaN}(\text{Not a Number}), & e = 255, f \neq 0 \end{cases}$$



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：-45.8125

- 绝对值45.8125的二进制数表示

- 00101101.1101  $\Rightarrow$  规格化  $1.011011101 \times 2^5 (V')$

- IEEE-754表示：

- 负数，符号位  $s=1$

- 阶/指数  $e=132$ ，二进制10000100（无符号整数）

$$\because E = e - bias(127) = 5 \therefore e = 5 + 127 = 132$$

- 尾数部分是  $V'$  小数点后的23位(不足补0)

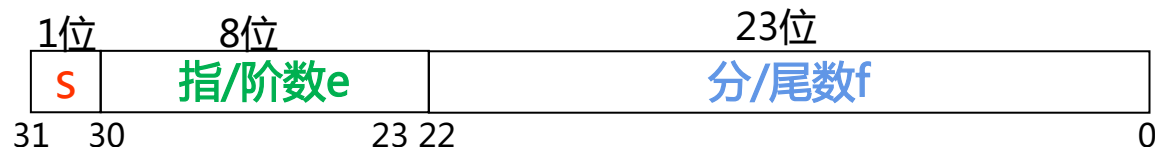
- 011011101000000000000000

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), e = 255, f = 0 \\ \text{NaN(Not a Number)}, e = 255, f \neq 0 \end{cases}$$

NaN(Not a Number),  $e = 255, f \neq 0$



1 10000100 0110 1110 1000 0000 0000 000



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：如下浮点数的十进制转换

0 01111010 000000000000000000000000

- 最高位  $s=0$ ，正数

- $e(01111010) = 122$

- $E = e - bias = 122 - 127 = -5$

- 尾数部分 000000000000000000000000

- 规格化表示为：

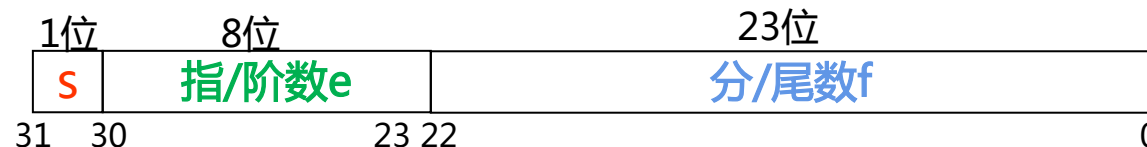
- $+1.000000000000000000000000 \times 2^{-5} \Rightarrow 1/32$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity} (+ \text{and } -), e = 255, f = 0$$

$$\text{NaN (Not a Number)}, e = 255, f \neq 0$$





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：如下浮点数的十进制转换

$V_{\text{float}} = \{$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), e = 255, f = 0$$

$$\text{NaN(Not a Number)}, e = 255, f \neq 0$$

0 1000101 111000011110000000000000

- 最高位  $s=0$ ，正数

- $e(10000101) = 133$

- $E = e - \text{bias} = 133 - 127 = 6$

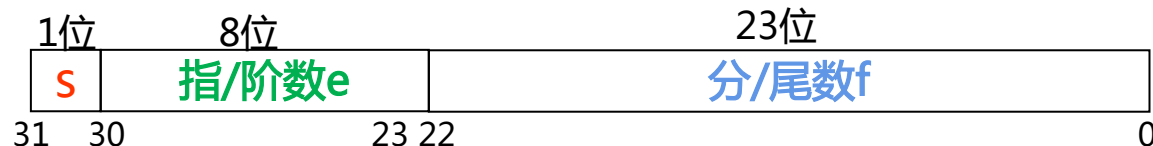
- $\text{frac} = 111000011110000000000000$

- 实际的尾数  $M = 1.111000011110000000000000$

- 规格化表示为：

- $+1.111000011110000000000000 \times 2^6 \Rightarrow 1111000.01111$

- 即120.46875





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：如下浮点数的十进制转换

$V_{\text{float}} = \{$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

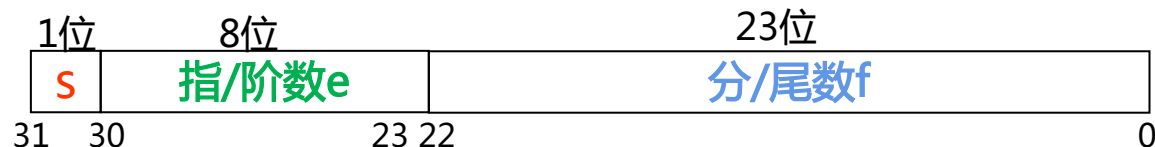
$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), e = 255, f = 0$$

$$\text{NaN(Not a Number)}, e = 255, f \neq 0$$

1 10000010 001010000000000000000000

- 最高位  $s=1$ ，负数
- $e(10000010) = 130$ 
  - $E = e - \text{bias} = 130 - 127 = 3$
- $\text{frac} = 001010000000000000000000$ 
  - 实际的尾数  $M = 1.001010000000000000000000$
- 规格化表示为：
  - $-1.001010000000000000000000 \times 2^3 \Rightarrow -1001.01$
  - 即 -9.25





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：如下浮点数的十进制转换

$V_{\text{float}} = \{$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

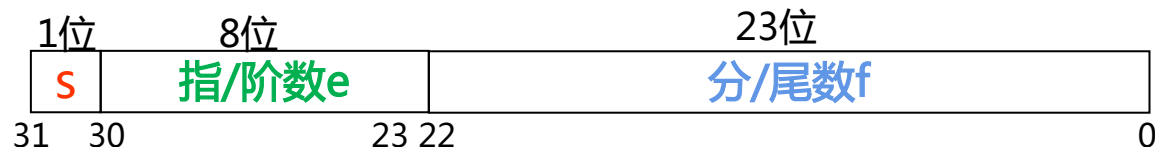
$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity} (+ \text{and } -), e = 255, f = 0$$

$$\text{NaN (Not a Number)}, e = 255, f \neq 0$$

0 11111110 111111111111111111111111

- 最高位  $s=0$ ，正数
- $e(11111110) = 254$
- $E = e - \text{bias} = 254 - 127 = 127$
- $\text{frac} = 111111111111111111111111$
- 实际的尾数  $M = 1.111111111111111111111111$
- 规格化表示为：
- $+1.111111111111111111111111 \times 2^{127} \Rightarrow \text{约等于 } 2^{128}$
- $1.111111111111111111111111 \text{ 约等于 } 2$





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

示例：如下浮点数的十进制转换

$V_{\text{float}} = \{$

$$(-1)^s \times 1.f \times 2^{e-127}, 1 \leq e \leq 254$$

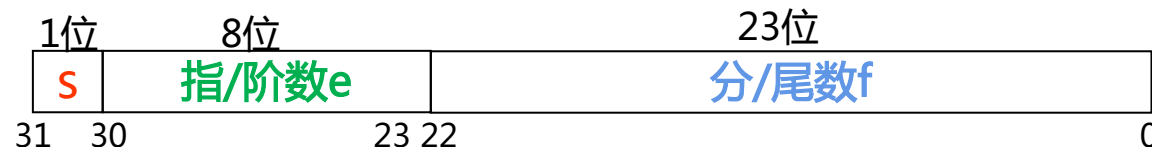
$$(-1)^s \times 0.f \times 2^{-126}, e = 0$$

$$(-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{ and } -), e = 255, f = 0$$

$$\text{NaN(Not a Number)}, e = 255, f \neq 0$$

1 00000000 000000000000000000000001

- 最高位  $s=1$ ，负数
- $e(00000000) = 0$
- $E = 1 - \text{bias} = 1 - 127 = -126$
- $\text{frac} = 00000000000000000000000001$
- 实际的尾数  $M = 0.00000000000000000000000001$
- 规格化表示为：
- $-0.00000000000000000000000001 \times 2^{-126} \Rightarrow -2^{-23} \times 2^{-126}$
- 即  $-2^{-149}$



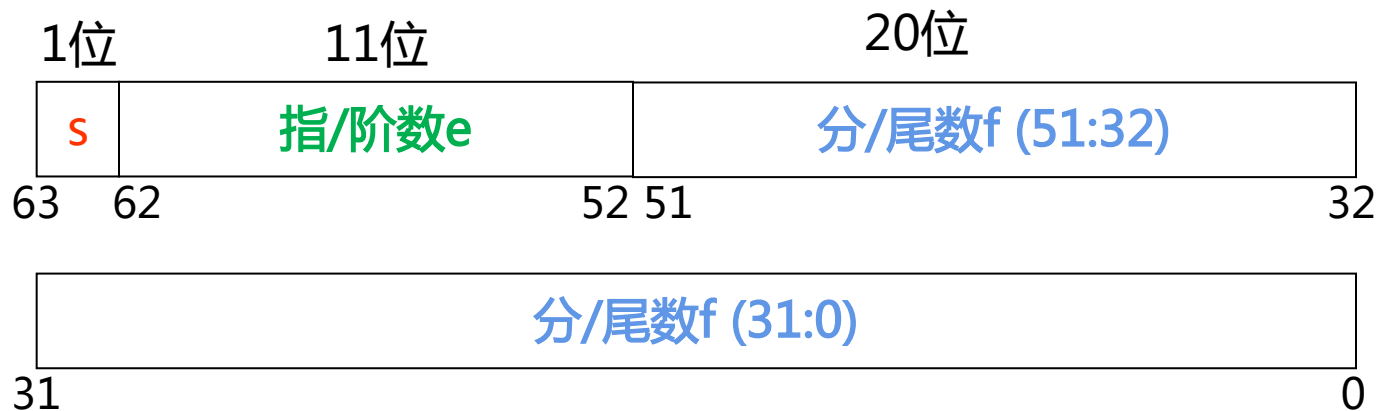


南京大學120周年校慶  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## IEEE-754双精度浮点数 ( double )

$$V = (-1)^S \times M \times 2^E$$



- 64位
- 符号 ( S )
- 0代表正数 , 1代表负数
- 指/阶数 ( e ) :
  - 无符号整数 ,  $[0, 2047]$  ( 11位 )
- 分/尾数 ( f ) :
  - 第0-51位 ( 52位 )
  - 精度约16位小数 ( 十进制数 )

$$V_{\text{double}} = \begin{cases} (-1)^s \times 1.f \times 2^{e-1023}, & 1 \leq e \leq 2046 \\ (-1)^s \times 0.f \times 2^{-1022}, & e = 0 \\ (-1)^s \times 1.f \times 2^{1024} = \text{Infinity}(+ \text{ and } -), & e = 2047, f = 0 \\ \text{NaN(Not a Number)}, & e = 2047, f \neq 0 \end{cases}$$





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### IEEE-754格式化浮点数表示范围

格式	最小正数	最大正数	最小负数	最大负数
单精度	$e = 1$ $f = 0$ $1.0 \times 2^{-126}$	$e = 254$ $f = 1 - 2^{-23}$ $(2 - 2^{-23}) \times 2^{127}$	$e = 254$ $f = 1 - 2^{-23}$ $-(2 - 2^{-23}) \times 2^{127}$	$e = 1$ $f = 0$ $-1.0 \times 2^{-126}$
双精度	$e = 1$ $f = 0$ $1.0 \times 2^{-1022}$	$e = 2046$ $f = 1 - 2^{-52}$ $(2 - 2^{-52}) \times 2^{1023}$	$e = 2046$ $f = 1 - 2^{-52}$ $-(2 - 2^{-52}) \times 2^{1023}$	$e = 1$ $f = 0$ $-1.0 \times 2^{-1022}$

注：均为真值



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 浮点数数据类型

### 课堂练习

- 转IEEE-754单精度浮点数（32位二进制表示）

- 32.9375

- $-32\frac{45}{128}$

- $-2^{-140}$

- 65536

$$V_{\text{float}} = \begin{cases} (-1)^s \times 1.f \times 2^{e-127}, & 1 \leq e \leq 254 \\ (-1)^s \times 0.f \times 2^{-126}, & e = 0 \\ (-1)^s \times 1.f \times 2^{128} = \text{Infinity}(+ \text{and } -), & e = 255, f = 0 \\ \text{NaN(Not a Number)}, & e = 255, f \neq 0 \end{cases}$$



南京大学120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022

## 02 ASCII编码



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## ASCII编码

### 如何表示从键盘输入计算机或者显示在显示器上的字符？

- ASCII ( American Standard Code for Information Interchange )
- 美国信息交换标准码, 美国国家标准局制定
- 8个二进制位表示
- 键盘上的每个键被一个唯一的ASCII码所识别
- 在键盘上敲击某个键时, 相应的8位码被存储, 并提供给计算机



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# ASCII编码

## ASCII

- 8位ASCII码的标准集
  - D列-字符对应的ASCII码的十进制表示
  - H列-十六进制表示
  - ASCII码字符的二进制？
- 大写字母A的ASCII码
  - 十进制D：65
  - 十六进制H：X41
  - 二进制B：01000001
- 小写字母a的ASCII码
  - 十进制D：97
  - 十六进制H：X61
  - 二进制B：01100001

字符	ASCII		字符	ASCII		字符	ASCII		字符	ASCII	
	D	H		D	H		D	H		D	H
NUL	0	00	SP	32	20	@	64	40	\	96	60
SOH	1	01	!	33	21	A	65	41	a	97	61
STX	2	02	"	34	22	B	66	42	b	98	62
ETX	3	03	#	35	23	C	67	43	c	99	63
EOT	4	04	\$	36	24	D	68	44	d	100	64
ENQ	5	05	%	37	25	E	69	45	e	101	65
ACK	6	06	&	38	26	F	70	46	f	102	66
BEL	7	07	'	39	27	G	71	47	g	103	67
BS	8	08	(	40	28	H	72	48	h	104	68
HT	9	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	5C		124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	^	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F	DEL	127	7F



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## ASCII编码

### ASCII

- 注意数字符号，大小写字母的位置和顺序
  - 按照顺序分配
  - 大小写英文字母的ASCII的差值均相同
- 这些规律可用于处理一些字符问题
  - 已知A的ASCII码值为65，推断a为97

$$D(a) - D(A) = 32$$

- 第0 ~ 32(20<sub>H</sub>)号及第127(7F<sub>H</sub>)号(共34个)是**控制字符**或**通讯专用字符**，其中第32号(20<sub>H</sub>)是空格(Space)
- 第33 ~ 126号(共94个)是**可见字符**，包括了阿拉伯数字，英文字母，英文标点等，可以通过标准键盘直接输入

字符	ASCII		字符	ASCII		字符	ASCII		字符	ASCII	
	D	H		D	H		D	H		D	H
NUL	0	00	SP	32	20	@	64	40	`	96	60
SOH	1	01	!	33	21	A	65	41	a	97	61
STX	2	02	"	34	22	B	66	42	b	98	62
ETX	3	03	#	35	23	C	67	43	c	99	63
EOT	4	04	\$	36	24	D	68	44	d	100	64
ENQ	5	05	%	37	25	E	69	45	e	101	65
ACK	6	06	&	38	26	F	70	46	f	102	66
BEL	7	07	'	39	27	G	71	47	g	103	67
BS	8	08	(	40	28	H	72	48	h	104	68
HT	9	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	5C		124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	^	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F	DEL	127	7F



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## ASCII编码

### ASCII

- 阿拉伯数字与其ASCII码之间的关系？
- 大写字母与相应的小写字母ASCII码之间的关系？
- 字母表顺序与ASCII码之间的关系？
- 其他字符？
  - <http://www.unicode.org/>
- ASCII码值的运算并没有新的运算，按照整数进行算术和逻辑运算

字符	ASCII		字符	ASCII		字符	ASCII		字符	ASCII	
	D	H		D	H		D	H		D	H
NUL	0	00	SP	32	20	@	64	40	`	96	60
SOH	1	01	!	33	21	A	65	41	a	97	61
STX	2	02	"	34	22	B	66	42	b	98	62
ETX	3	03	#	35	23	C	67	43	c	99	63
EOT	4	04	\$	36	24	D	68	44	d	100	64
ENQ	5	05	%	37	25	E	69	45	e	101	65
ACK	6	06	&	38	26	F	70	46	f	102	66
BEL	7	07	'	39	27	G	71	47	g	103	67
BS	8	08	(	40	28	H	72	48	h	104	68
HT	9	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	5C		124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	^	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F	DEL	127	7F



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## 其他数据类型

- 字符串

- 字符序列，以 NUL (0)结束

- 汉字字符

- 一个字就是一个方块图形
- 如GB2312-80字符集
- 超过6万个字（至少16位）

- 图像

- 像素矩阵
- 黑白: 1位 (1/0 = black/white)
- 彩色: Red, Green, Blue (RGB) （每个8 位）
- (0,0,0) Black, (255,0,0) Red, (255,255,255) White
- 其他属性: 透明度





南京大学120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022

## 03 C语言中的数据类型



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 深入理解C语言中的数据类型

- char , ASCII码
- int , 二进制补码整数类型
- float , 单精度浮点数
- double , 双精度浮点数
- 每种类型采用多少位二进制来表示 , 与具体计算机的指令集结构和编译器有关



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

C语言		字节	
有符号	无符号	32位字长机器	64位字长机器
字符型 [signed] char	字符型 unsigned char	1	1
短整型 short	短整型 unsigned short	2	2
整型 int	整型 unsigned int	4	4
长整型 long	长整型 unsigned long	4	8
整型 ( 32位 ) int32_t	整型 ( 32位 ) uint32_t	4	4
整型 ( 64位 ) int64_t	整型 ( 64位 ) uint64_t	8	8
单精度浮点型 float		4	4
双精度浮点型 double		8	8



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

C语言	字节	最小值	最大值
[signed] char	1	$-2^7$	$2^7-1$
unsigned char	1	0	$2^8-1$
short	2	$-2^{15}$	$2^{15}-1$
unsigned short	2	0	$2^{16}-1$
int	4	$-2^{31}$	$2^{31}-1$
unsigned int	4	0	$2^{32}-1$
long	8	$-2^{63}$	$2^{63}-1$
unsigned long	8	0	$2^{64}-1$
int_32	4	$-2^{31}$	$2^{31}-1$
uint32_t	4	0	$2^{32}-1$
int64_t	8	$-2^{63}$	$2^{63}-1$
uint64_t	8	0	$2^{64}-1$
float	4		
double	8		



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

### char ( 字符型 , Character Type )

- **表示字符**：如字母 ( A、b、C )、数字 ( 0、1、2 )、标点符号 ( .、!、? ) 等
- **字符编码**：每个字符在计算机内部都有一个与之关联的唯一数字编码，如ASCII
- **文本处理**：处理文本、字符串和字符数组，如连接、截取、搜索、替换等
- **输入和输出**：允许程序与用户进行文本交互，如命令行界面或文本文件的读写
- **字符常量和变量**：表示字符值，或定义字符型变量，用于存储和处理字符数据

```
char grade = 'A'; // 声明一个字符型变量并初始化为字符'A'
```

```
char newline = '\n'; // 可以用转义序列来表示特殊字符，如换行符
```



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

### char ( 字符型 , Character Type )

- char 类型采用ASCII码表示
- 整数运算
- 如果key是char类型的变量：
  - $((\text{'a'} \leq \text{key}) \ \&\& \ (\text{key} \leq \text{'z'})) \ || \ ((\text{'A'} \leq \text{key}) \ \&\& \ (\text{key} \leq \text{'Z'}))$
  - 表达式 “ ‘a’ <= key ” 中的 “<= ” 运算符比较的就是变量key和字符a的ASCII码的大小



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

### int ( 整型 , Integer )

- **表示整数**：包括正整数、负整数和零
- 存储和处理整数值，如计数、索引、循环计数器等
- 大小和范围可以因计算机体系结构和编程语言而异
  - **int ( 整型 )**
    - 4个字节 ( 32位 ) 内存空间，范围约 $-2^{31}$ — $2^{31}-1$
  - **short int ( 短整型 )**
    - 2个字节 ( 16位 ) 内存空间，范围约 $-2^{15}$ — $2^{15}-1$
  - **long int ( 长整型 )**
    - 8个字节 ( 64位 ) 内存空间，范围约 $-2^{63}$ — $2^{63}-1$
  - .....
- 声明整型变量并对其进行赋值，也可以进行整数运算，如加减乘除等。

```
int age = 25; // 声明一个整型变量并初始化
```

```
int sum = 10 + 20; // 整数运算，将10和20相加并将结果存储在sum变量中
```



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



## C语言中的数据类型

### float , double ( 单精度浮点数和双精度浮点数 )

- 不同型之间进行强制类型转换，结果可能会超出预期（溢出/舍入）
  - int类型->float类型：可能被舍入，不会溢出
    - 单精度浮点的尾数字段是23位，可能无法保留精度
  - int类型/float类型->double类型：可保留精度数值
    - double类型具有更大的范围和精度
  - 从double类型->float类型：可能溢出或被舍入
    - float类型所表示的数值范围和精度更小
  - 当float类型/double类型->int类型：可能被舍入或溢出
    - 向0舍入，例如1.9将会被转换成1，-1.9将被转换成-1
    - 或发生溢出

#### 溢出 ( Overflow )

溢出发生在转换超出目标类型所能表示的范围。

#### 舍入 ( Rounding )

舍入发生在转换过程中丢失精度时。





南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 混合类型表达式的运算

- 算术运算表达式
  - “ $i + 3.1$ ”，其中“ $i$ ”被声明为int类型，3.1是浮点型字面常量
  - 将整数转换为浮点数，然后进行计算（隐式的数据类型转换）
- 整数与字符型运算
  - “ $x + 'a'$ ”，其中“ $x$ ”为int类型，字符'a'的ASCII码为97
  - 字符型将被转化为整数类型后再进行计算
  - 如果int类型变量x取值为1，则计算 “ $1 + 97$ ”，表达式的值为98



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 输入输出的格式说明

- 格式说明 “%d”
- 输出
  - 它使得列在格式用字符串后面的数值被显示为十进制数输出
  - 事实上，是将一个存储的二进制数转化为ASCII码字符输出
  - 例如，语句 “printf (“25 plus 76 in decimal is %d. \n”, 25 + 76)”
  - 输出 “25 plus 76 in decimal is 101.”
- 输入
  - 它将输入的数据解释为十进制数值
  - 事实上，是将输入字符的ASCII码转化为二进制补码整数存储



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 输入输出的格式说明

- 格式说明 “%x” ( 八进制的 “%o” 同理 )
- 输出
  - 它使得列在格式用字符串后面的数值被显示为**十六进制数**
  - 语句 “printf (“25 plus 76 in hexadecimal is %x. \n”, 25 + 76)”
  - 输出 “25 plus 76 in hexadecimal is 65.”
- 输入
  - 它将输入的数据解释为十六进制数值
  - 事实上，将输入字符的**ASCII码转化为二进制补码整数存储**
  - scanf ("%x", &valueX);
  - 输入： A , A0 , a , ae , 10 , 0XA , 0x10
  - 存储： 10 , 160 , 10 , 174 , 16 , 10 , 16



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 输入输出的格式说明

### ➤ 格式说明 `"%c"`

### ➤ 输出

- 它使得列在格式用字符串后面的数值被解释为ASCII字符显示
- 语句 `"printf ("25 plus 76 as a character is %c. \n", 25 + 76)"`
- 输出 `"25 plus 76 as a character is e."`

### ➤ 输入

- `scanf ("%c", &grade);`
- 事实上，这个过程是将输入的字符的ASCII码进行存储



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



# C语言中的数据类型

## 输入输出的格式说明

### ➤ 格式说明 `"%c"`

### ➤ 输出

- 它使得列在格式用字符串后面的数值被解释为ASCII字符显示
- 语句 `"printf ("25 plus 76 as a character is %c. \n", 25 + 76)"`
- 输出 `"25 plus 76 as a character is e."`

### ➤ 输入

- `scanf ("%c", &grade);`
- 事实上，这个过程是将输入字符的ASCII码进行存储



南京大學120周年校庆  
120<sup>th</sup> ANNIVERSARY  
NANJING UNIVERSITY  
1902 - 2022



习题

- 书面作业
  - 6.10
  - 6.11
  - 6.12
  - 6.13
  - 6.14
  - 6.15
  - 6.17



# 谢 谢

诚耀百廿 雄创一流

