Types, Variables, Operators and Expressions (I)

Lecture 02

Min Zhang

zhangmin@sei.ecnu.edu.cn

2020.09.28

Software Engineering Institute

10:00-11:40, Monday, Room 319
Software Engineering Institute, East China Normal University

Contents of today's lecture

- Types
- Variables
- Operators
- Expression

Type

物以类聚，人以群分！

A quiz

What are the types of following items:

0, 1, 2, 3, 4	**int**
1.2, 2.0, 3.5	**float**
’0’, ’1’, ’2’, ’3’, ’a’, ’b’	**char**
”0”, ”1”, ”2”, ”01”, ”a”, ”ab”	**string**
’abc’	**bad type**

What is type?

Definition (Type)

A type is a name for a class (set) of something!

- **int**: a class (set) of integer numbers
  (Note: not all the integer numbers, why?)

- **float**: a class of float numbers
  (Note: not all the float numbers, why?)

- **char**: THE class of ALL the characters
  (Note: this time all, how many?)

- **string**: the set of arrays of characters
  No string type in C language.

Constants

Definition (Constant)

A constant is an element in a set of items.

Example

0, 1, 2, 3, 4	constants of **int**
1.2, 2.0, 3.5	constants of **float**
'0', '1', '2', '3', 'a', 'b'	constants of **char**
"0", "1", "2", "01", "a", "ab"	constants of **string**

How data are represented in computers?

In computers, there are only two values: 0 and 1.

Question: How to represent all the data such as integers, float numbers, and characters?

We invent new approach

1. integer 0: 0
2. integer 1: 1
3. integer 2: 10
4. integer 3: 11
5. integer 4: 100
6. integer 5: 101
7. integer 6: 110
8. integer 7: 111
9. integer 8: 1000
10. integer 9: 1001
11. integer 255: 11111111

Bit and byte

Definition (Bit and byte)

A bit is one 0 or 1 in computer, and a byte is 8 bits.

Bit: **位**
Byte: **字节**

What is the range of integer numbers which can be represented by 4 byte (32 bit)?

$0 \sim 2^{32} - 1 \quad (2^{32} = 4,294,967,296)$

or, $-(2^{31}) \sim 2^{31} - 1$

Here is a problem

Question: How to represent negative integers in computer?

Example (1 byte case)

11111111B = 255

11111111B = -127 (true form, 原码)

11111111B = -0 (one's complement, 一的补数)

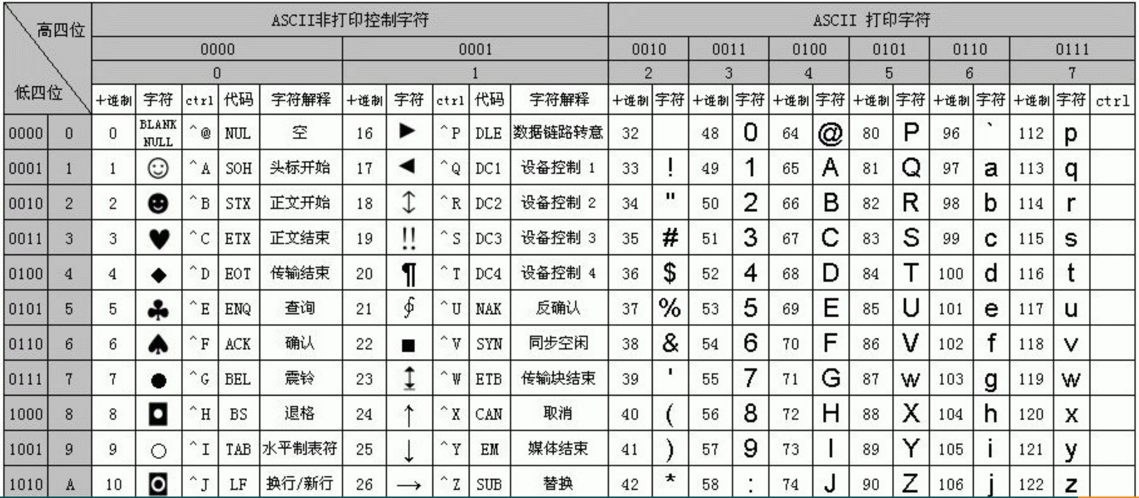
11111111B = -1 (two's complement, 二补数，补码)

思考：补码为什么是原码取反加 1？

Click here for more details.

Here is another problem

Question: How to represent characeters in computer?



高四位		ASCII非打印控制字符										ASCII 打印字符												
		0000					0001					0010		0011		0100		0101		0110		0111		
		0					1					2		3		4		5		6		7		
低四位		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	►	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH	头标开始	17	◄	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	◆	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	●	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	

Solution: to use 7 bits

Please remember:

1. '0'-'9': ASCII values 48∼57
2. 'a'-'z': ASCII values 97∼122
3. 'A'-'Z': ASCII values 65∼90

Here is another problem

What does **00110000** represent in a computer?

1. an integer 48
2. a character ’0’

We have ’0’==48 C program!!!

But, 0!=48

Type again

Basic types:

```
int       // 4 bytes integer
char      // 1 byte
float     // 4 bytes float numbers
double    // 8 bytes float numbers
```

Type adjectives:

```
short
long
signed
unsigned
```

Complex types:

```
short int      // 2 bytes integer
unsigned int   // non-negative integer with 4 bytes
long unsigned int // non-negative integer with 8 bytes
```

The length of byte may be different in different compiling environment.

Variables

Definition (Variable)

A variable is something which has a type and whose value can be modified.

A variable has

1. A name
2. A type: what kind of values can be assigned to it
3. A value
4. An address: where the value is stored

Variable declaration and definition

```c
int i;   // declare a variable i of type int
float i,j; // declare two variables i and j of type float
char ch='a'; // declare a character variable ch and assign 'a' to it
char ch1='b',ch2='c'; // declare two character variables and assign
                   // 'b' and 'c' to them, respectively
const double e = 2.71828182845905; // variable e cannot be modified
```

Note:

- remember initializing your variables before using them
- do not use keywords as variable names
- x and X are different variables

Arithmetic operators

- +: 5+2, result: 7
- -: 5-2, result: 3
- *: 5*2, result: 10
- /: 5/2, result: 2 not 2.5
- %: 5%2, result: 1

Remember:

The type of the result is ALWAYS THE SAME AS the operation of its parameters.

// 黄金定律：结果的类型一定和根运算符的类型一致

Relational operators and logical operators

Relations operators:

- >: `'0'>0`, result: 1
- >=: `'0'>=0`, result: 1
- <: `'0'<0`, result: 0
- <=: `'0'<=0+48`, result: 1
- == : `'0'<=0+48`, result: 1
- != : `'0'!=0+48`, result: 0

Logical operators:

- && : `'0'<0+48 && '1'==1+48`, result: 0
- || : `'0'<0+48 || '1'==1+48`, result: 1

Remarks

Remember:

The result of logical operation is only 1 or 0

Example: `int i='0'-48==0;`, value of `i`: 1

The privilege of logical operators is lower than arithmetic ones

Example: `int i='0'-48==0+1;`, value of `i`: 0

Type conversion

- Automatic conversion: from small-size type to big-size type

  Example: `5/2.0`, result: `2.5`. Here, `5` → `5.0`

- Compulsory Type Conversion: to force the conversion from a type to another one

  Example:

  ```
  float pi=3.1415926;
  int pi2 = (int) pi;
  ```

  The value of `pi2`: 3

Summary

Today's topics:

1. Data representation in computer
2. Types
3. Constants
4. Variables
5. Operators
6. Type conversion

Homework

- Learn how to use `printf` to print different types of data.
  Read this article for more details.

- Try writing programs to solve the problems 1073,1147,1828 on OJ.

What's coming next?

1. Bit operators
2. Privilege of operators
3. ++,–
4. Assignment operator
5. Conditional expression