|  |  |
| --- | --- |
| 1. bitset类型 2. 定义和初始化bitset 3. 用unsigned值初始化bitset 4. 从一个string初始化bitset | Bitset<32> bitvec(1U);  低位，高位  Bitset<13> bitvec1(0xbeef);  Bitset<32> bitvec5(str,5,4);  Bitset<32> bitvec6(str,str.size()-4);  Bitset<32> bitvec4(“1100”); |
| 1. bitset类型 2. 使得位运算的使用更为容易，并且能够处理超过最长整形类型大小的位集合 3. 定义和初始化bitset 4. 是一个类模板，具有固定大小，包含多少个二进制位，大小必须是一个常量表达 5. bitset中的二进制位也是未命名的，我们通过位置来访问他们。 6. 从0开始编号。从0到31的32个二进制位。编号从0开始的二进制位被称为低位，编号到31结束的二进制位被称为高位 7. 用unsigned值初始化bitset 8. 使用一个整形值来初始化bitset时，此值将被转换为unsigned long long类型并被当做位模式来处理。 9. 若bitset的大小大于一个unsigned long long中的二进制位数，则剩余的高位为0，   若bitset的小小小于一个unsigned long long中的二进制位数，则只使用给定值中的低位，超出bitset大小的高位被丢弃   1. 在64位机器中，long long 0ULL是64个0比特   Bitset<128> bitvec3(~0ULL);   1. 从一个string初始化bitset 2. 若string包含的字符数比bitset少，则bitset的高位为0 3. String的下标编号与bitset恰好相反，string下标最大的字符对应低位   最小的字符对应高位   1. 可以不用整个string，可以用一个子串 | |



