

**Ordenamiento por intercambio de residuos:** es útil si las claves son suficientemente grandes.

```
class clavebits
{ private: int x;
  public:
    clavebits& operator=(int i)
    { x=i; return *this; }
    inline unsigned bits (int k, int j)
    { return (x>>k)&~(~0<<j)}
};
typedef clavebits tipoElemento;

void cambioresiduos(tipoElemento a[], int izq, int der, int b)
{ int i,j; tipoElemento t;
  if (der>izq && b>0)
  { i= izq; j=der;
    while(j!=i)
    { while(!a[i].bits(b,1) && i<j)i++;
      while( a[j].bits(b,1) && j>i)j--;
      intercambio(a, i, j);
    }
    if (!a[i].bits(b,1)) j++;
    cambioresiduos(a, izq, j-1, b-1);
    cambioresiduos(a, j, der, b-1);
  }
}
```

	Clave	código
1	E	00101
2	J	01010
3	E	00101
4	M	01101
5	P	10000
6	L	01100
7	O	01111
8	A	00001
9	O	01111
10	R	10010
11	D	00100
12	E	00101
13	N	01110
14	A	00001
15	R	10010