Counting sort

Counting sort is a sorting technique based on keys between a specific range. It works by counting the number of objects having distinct key values (kind of hashing).

For simplicity, consider the data in the range 0 to 9.

1	4	1	2	7	5	2

Take a count array *m* to store the count of each unique object. For each value of *x* from input array make m[*x*]++. We have:

- two one's (m[1]++ twice)
- two two's (m[2]++ twice)
- one four, five and seven (m[4]++; m[5]++; m[7]++)

	0	1	2	3	4	5	6	7	8	9
m	0	2	2	0	1	1	0	1	0	0

E-OLYMP **2327. Counting sort** Use the magic of counting sort: sort *n* numbers in the range [0; 100000].

► Input numbers can be sorted using counting sort, as they are integers and we know their range.

Input numbers store in array m.

```
#define MAX 100010
int m[MAX];
```

Read the input data. In m[*i*] count the number of times the value *i* is encountered among the given numbers.

```
scanf("%d",&n);
memset(m,0,sizeof(m));
for(i = 0; i < n; i++)
{
  scanf("%d",&v);
  m[v]++;
}
```

Print the resulting array. Number *i* must be printed m[*i*] times.

```
for(i = 0; i < MAX; i++)
  for(j = 0; j < m[i]; j++)
    printf("%d ",i);
printf("\n");
```

E-OLYMP **2617. Birthdays** If somebody in Summer Computer School (SCS) has a birthday, the celebration is arranged. Help organizers to find out how many times they must arrange the birthday parties during SCS.August.

► Note that we are only interested in birthdays for August (one month), that has 31 days. For each day from 1 to 31, calculate how many LKSH students were born on that day. And then count the number of days when LKSH students celebrate birthdays.

E-OLYMP **354. Permutation** The sequence of *n* positive integers is given. Determine whether the sequence is a permutation of the first *n* positive integers.

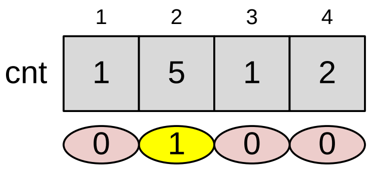
► Declare the linear array m of length $n \leq 10000$. Put the amount of numbers *i* in the input sequence into cell m[*i*]. If all values m[1], m[2], …, m[*n*] are nonzero (there are exactly *n* input numbers), then all these values are equal to 1 and input seuence is a permutation. Otherwise print the smallest *i* for which m[*i*] is zero.

If some number of the input sequence is greater than *n*, then such sequence is not a permutation.

E-OLYMP **145. Squares** Given the length of n segments. What is the maximum number of squares can be constructed? Each side of a square must be constructed from only one segment.

► Let we have *k* segments of the same length. Then you can make *k* / 4 squares out of them. The lengths of the segments vary from 1 to 100. Count the number of segments of length *i* ($1 \leq i \leq 100$) in cnt[*i*]. Then the maximum possible number of squares that can be made out of the given segments is (cnt[1] + cnt[2] + …. + cnt[100]) / 4.

Consider the state of the *cnt* array after counting sort.



	1	2	3	4
cnt	1	5	1	2
	0	1	0	0

From segments of length 2, you can make 5 / 4 = 1 square. It is impossible to make squares out of the remaining lengths of the line segments.

E-OLYMP **7809. Morning exercises** In the morning, many students do their exercises. According to established tradition, the students dance in the branded T-shirts. During the first three days of the camp, schoolchildren and teachers noticed that a couple who dances in identical T-shirts looks more aesthetically better. Therefore before the exercises, they decided first to put pairs of children in identical T-shirts, and then the remaining ones. The excellent student Seryozha wanted to know what the largest number of aesthetic pairs can be formed from everyone who came to morning exercises.

► T-shirt color is a number from 0 to 9. Let’s count the number of T-shirts of color *i* in m[*i*]. Then, out of schoolchildren wearing a T-shirt of color *i*, one can make m[*i*] / 2 aesthetic pairs. Then the total number of aesthetic pairs is

$$(m[0] + m[1] + \ldots + m[9]) / 2$$