

1. 【NOIP2010】

```
01 #include<iostream>
02 using namespace std;
03 void swap(int &a,int &b)
04 {
05     int t;
06     t=a;
07     a=b;
08     b=t;
09 }
10 int main()
11 {
12     int a1,a2,a3,x;
13     cin>>a1>>a2>>a3;
14     if (a1>a2)
15         swap(a1,a2);
16     if (a2>a3)
17         swap(a2,a3);
18     if (a1>a2)
19         swap(a1,a2);
20     cin>>x;
21     if (x<a2)
22         if (x<a1)
23             cout<<x<<' '<<a1<<' '<<a2<<' '<<a3<<endl;
24         else
25             cout<<a1<<' '<<x<<' '<<a2<<' '<<a3<<endl;
26     else
27         if (x<a3)
28             cout<<a1<<' '<<a2<<' '<<x<<' '<<a3<<endl;
29         else
30             cout<<a1<<' '<<a2<<' '<<a3<<' '<<x<<endl;
31     return 0;
32 }
```

●判断题

- (1) 去掉第 14 行到第 19 行,不会影响程序的运行结果。 ()
- (2) 将第一行的 `iostream` 改成 `cstdio` 会编译错误。 ()
- (3) 本题输出结果和将四个数从小到大排序一样。 ()
- (4) 如果输入 `91 2 20\n77`,输出为 `2 20 77 91`。 ()

●选择题

- (5) 如果输入 `114514 191 810\n258`,程序输出()。

A. 114 514 258 1919810

B. 114514 191 810\n258

C. 114514 191 810 258

D. 191 258 810 114514

- (6) 如果输入 `1 1 1\n2`,程序输出()。

A. 1 1 1 2

B. 1 1 2 1

C. 1 2 1 1

D. 2 1 1 1

2. 【NOIP2010】

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=100;
06     int na,nb,a[SIZE],b[SIZE],i,j,k;
07     cin>>na;
08     for (i=1;i<=na;i++)
09         cin>>a[i];
10     cin>>nb;
11     for (i=1;i<=nb;i++)
12         cin>>b[i];
13     i=1;
14     j=1;
15     while ((i<=na)&&(j<=nb)) {
16         if (a[i]<=b[j]) {
17             cout<<a[i]<<' ';
18             i++;
19         }
20         else {
21             cout<<b[j]<<' ';
22             j++;
23         }
24     }
25     if (i<=na)
26         for (k=i;k<=na;k++)
27             cout<<a[k]<<' ';
28     if (j<=nb)
29         for (k=j;k<=nb;k++)
30             cout<<b[k]<<' ';
31     return 0;
32 }
```

●判断题

- (1) 保证 a 数组和 b 数组有序, 输出的序列一定是一个不降序列。 ()
- (2) 如果输入 0 0, 不会输出数。 ()
- (3) 如果删掉第 13 行和第 14 行不影响程序结果。 ()
- (4) 使用 C++98 不会 CE。 ()

●选择题

- (5) 该程序时间复杂度是()。
- A. $O(na+nb)$ B. $O(\max\{na,nb\}\log\max\{na,nb\})$
- C. $O(na \cdot nb)$ D. $O(na^{nb})$
- (6) 如果输入 5\n1 3 5 7 9\n4\n2 6 10 14, 输出()。
- A. 1 2 3 5 6 7 9 10 14 B. 14 10 9 7 6 5 3 2 1
- C. 1 3 5 7 9 2 6 10 14 D. 5 1 3 5 7 9 4 2 6 10 14

3. 【NOIP2011】

```
01 #include<iostream>
02 #include<cstring>
03 using namespace std;
04 const int SIZE=100;
05 int main(){
06     int n,i,sum,x,a[SIZE];
07     cin>>n;
08     memset(a,0,sizeof(a));
09     for (i=1;i<=n;i++){
10         cin>>x;
11         a[x]++;
12     }
13     i=0;sum=0;
14     while (sum<(n/2+1)){
15         i++;
16         sum+=a[i];
17     }
18     cout<<i<<endl;
19     return 0;
20 }
```

●判断题

- (1)当第8行的 `sizeof(a)` 改为 `SIZE` 时,运行结果不会发生改变。 ()
- (2)当第11行的 `a[x]++` 改成 `++a[x]` 时,运行结果不会发生改变。 ()
- (3)当第14行的 `sum<(n/2+1)` 改为 `sum<(n+1)/2` 时,运行结果不会发生改变。 ()
- (4)数组 `a` 的所有数中的最大值为1。 ()

●选择题

- (5)输入 11 4 5 6 6 4 3 3 2 3 2 1,输出的结果为()。
- A. 3 B. 4 C. 5 D. error
- (6)输入 5 1 2 3 4 5,输出的结果为()。
- A. 2 B. 3 C. 4 D. 5

4.【NOIP2014】

```
01 #include<iostream>
02 #include<string>
03 using namespace std;
04 const int SIZE=100;
05 int main() {
06     string dict[SIZE];
07     int rank[SIZE];
08     int ind[SIZE];
09     int i, j, n, tmp;
10     cin>>n;
11     for (i=1; i<=n; i++) {
12         rank[i]=i;
13         ind[i]=i;
14         cin>>dict[i];
15     }
16     for (i=1; i<n; i++)
17         for (j=1; j<=n-i; j++)
18             if (dict[ind[j]]>dict[ind[j+1]]) {
19                 tmp=ind[j];
20                 ind[j]=ind[j+1];
21                 ind[j+1]=tmp;
22             }
23     for (i=1; i<=n; i++)
24         rank[ind[i]]=i;
25     for (i=1; i<=n; i++)
26         cout<<rank[i]<<" ";
27     cout<<endl;
28     return 0;
29 }
```

●判断题

- (1)该程序的本质是按字典序对字符串排序。 ()
- (2)如果输入 0,没有任何输出。 ()
- (3)如果使用 C++98 编译,不会出现编译错误。 ()
- (4)如果去掉第 12 行,不影响。 ()

●选择题

(5)输入 7\naaaa\nnaba\nnbba\nnaaa\nnaaa\ncccc\nnaa,输出()。

- A. 2 5 6 3 4 7 1
- C. 1 7 4 3 6 5 2

- B. 1 2 3 4 5 6 7
- D. 7 6 5 4 3 2 1

(6)这个程序使用的是什么排序()。

- A. 插入排序
- B. 希尔排序
- C. 选择排序
- D. 冒泡排序

5. 【NOIP2016】

```

01#include<iostream>
02 using namespace std;
03 int main() {
04     int a[6];
05     int pi=0;
06     int pj=5;
07     int t,i;
08     while (pi<pj){
09         t=a[pi];
10         a[pi]=a[pj];
11         a[pj]=t;
12         pi++;
13         pj--;
14     }
15     for (i=0;i<6;i++)
16         cout<<a[i]<<" ";
17     cout<<endl;
18     return 0;
19 }

```

●判断题

- (1)将第 8 行改成 $pi \leq pj$ 不影响程序结果。 ()
- (2)程序输出六个数,逗号只出现在相邻两个数之间。 ()
- (3)如果 $a=\{1,2,3,4,5,6\}$,那么输出 6,5,4,3,2,1。 ()

●选择题

- (4)这个程序在()。
- A. 将已知序列翻转 B. 求序列的逆
- C. 求序列的卷积 D. 将原始数列随机打乱
- (5)这个程序的时间复杂度是()。
- A. $O(n^2)$ B. $O(n)$ C. $O(n \log n)$ D. $O(n\sqrt{n})$
- (6)如果 $a=\{6,5,4,3,2,1\}$,那么输出()。
- A. 6,5,4,3,2,1 B. 6 5 4 3 2 1 C. 1 2 3 4 5 6 D. 1,2,3,4,5,6,

6. 【NOIP2017】

```
01 #include<iostream>
02 using namespace std;
03 int n,s,a[100005],t[100005],i;
04 void mergesort(int l,int r){
05     if (l==r)
06         return;
07     int mid=(l+r)/2;
08     in tp=l;
09     int i=l;
10     int j=mid+1;
11     mergesort(l,mid);
12     mergesort(mid+1,r);
13     while (i<=mid&& j<=r){
14         if (a[j]<a[i]){
15             s+=mid-i+1;
16             t[p]=a[j];
17             p++;
18             j++;
19         }else {
20             t[p]=a[i];
21             p++;
22             i++;
23         }
24     }
25     while (i<=mid){
26         t[p]=a[i];
27         p++;
28         i++;
29     }
30     while (j<=r){
31         t[p]=a[j];
32         p++;
33         j++;
34     }
```



```

34     }
35     for (i=1;i<=r;i++)
36         a[i]=t[i];
37 }
38
39 int main() {
40     cin>>n;
41     for (i=1;i<=n;i++)
42         cin>>a[i];
43     mergesort(1,n);
44     cout<<s<<endl;
45     return 0;
46 }
47

```

●判断题

- (1)如果将第 10 行的 $\text{mid}+1$ 改成 mid ,不会影响程序结果和时间复杂度。 ()
- (2)如果将第 7 行的 $(l+r)/2$ 改成 $(l+r+1)/2$ 不会影响程序结果和时间复杂度。 ()
- (3)程序输出结果不可能是 0。 ()

●选择题

- (4)该程序的时间复杂度是()。

A. $O(n)$ B. $O(n\log n)$ C. $O(n\sqrt{n})$ D. $O(\frac{n^2}{w})$

- (5)如果输入 6 2 6 3 4 5 1,输出()。

A. 1 B. 2 C. 4 D. 8

- (6)该程序是求输入序列的()。

A. 元素总和 B. 逆序列 C. 逆序对 D. 卷积