

2024 蓝桥杯选拔赛题解

2024 蓝桥杯选拔赛题解

第一题

第二题

第三题

第四题

第五题

第六题

第七题

第八题

第九题

第十题

文档制作：漂至

75min交卷，没有检查代码，可能有误，仅供参考

第一题

- 统计质因数，直接枚举就可以了

```
1 #include<bits/stdc++.h>
2 #define x first
3 #define y second
4
5 using namespace std;
6
7 typedef long long LL;
8 typedef pair<int,int> PII;
9
10 /*
11     解题思路：
12
13     */
14
15 const int N=1e5+5;
16
```

```

17 int n=2024;
18
19 bool is_prime(int x) {
20     for(int i=2;i<=x/i;i++) {
21         if(x%i==0) return false;
22     }
23     return true; // 质数
24 }
25
26 int cnt;
27
28 int main() {
29     // 1不是质数
30     for(int i=2;i<=n;i++) {
31         if(is_prime(i) && n%i==0) {
32             cout<<i<<'\\n';
33             cnt++;
34         }
35     }
36     cout<<cnt;
37     return 0;
38 }
39
40 /*
41     输入样例:
42
43     输出样例:
44
45 */

```

第二题

- 开根地板除

```
1 #include<bits/stdc++.h>
2 #define x first
3 #define y second
4
5 using namespace std;
6
7 typedef long long LL;
8 typedef pair<int,int> PII;
9
10 /*
11     解题思路:
12
13     */
14
15 const int N=1e5+5;
16
17 int n,a[N];
18 int cnt;
19
20 int main() {
21     n=2024;
22     while(n!=1) {
23         cnt++;
24         n=floor(sqrt(n));
25     }
26     cout<<cnt;
27     return 0;
28 }
29
30 /*
31     输入样例:
32
33     输出样例:
34
35     */
```

第三题

- 找距离2024最近的平方根

```
1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4
5  using namespace std;
6
7  typedef long long LL;
8  typedef pair<int,int> PII;
9
10 /*
11     解题思路:
12
13     */
14
15 const int N=1e5+5;
16
17 int n,a[N];
18
19 int main() {
20     n=2024;
21     for(int i=1;i<=10000;i++) {
22         if(i*i*i>=2024) {
23             cout<<i-2024;
24             break;
25         }
26     }
27     return 0;
28 }
29
30 /*
```

```
31  输入样例：
32
33  输出样例：
34
35  */
```

第四题

- 日期统计，稍麻烦一点，思路：枚举日期用字符串存储，像月和日不足两位则补0，需要考虑闰年，每月多少天用数组表示，计算和19010101的天数差值，通过%运算计算星期几

```
1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4
5  using namespace std;
6
7  typedef long long LL;
8  typedef pair<int,int> PII;
9
10 /*
11     解题思路：用字符串枚举年月日
12
13     */
14
15 const int N=1e5+5;
16 int Months[]=
17     {0,31,28,31,30,31,30,31,31,30,31,30,31}; // 平年
18 const string weekday[]={"星期一","星期二","星期三",
19     "星期四","星期五","星期六","星期天"};
20 string s;
21
22 int cnt; // 好日期
```

```
21
22 int main() {
23     int diff=0; // 计算相差多少天
24     // 年
25     for(int year=1901;year<=2024;year++) {
26         s+=to_string(year);
27         // 如果是闰年，把Months[2]改成29
28         if(year%4==0 && year%100!=0 ||
year%400==0) Months[2]=29;
29         // 月
30         for(int i=1;i<=12;i++) {
31             if(!(i/10)) s+="0"; // 不足10
32             s+=to_string(i);
33             // 日
34             for(int j=1;j<=Months[i];j++) {
35                 if(!(j/10)) s+="0";
36                 s+=to_string(j);
37             // cout<<s<<' ';
38
39             // 计算差日
40             diff+=0;
41             int sum=0;
42             // 年
43             for(int x=1;x<=12;x++)
sum+=Months[x];
44             diff+=(year-1901)*sum;
45             // 月
46             for(int x=1;x<=i-1;x++)
diff+=Months[x];
47             // 日
48             diff+=j;
49             // 计算星期几
50             // cout<<diff<<' ';
51             // cout<<weekday[diff%7]<<'\n';
52
```

```

53         if(weekday[diff%7]=="星期一" &&
s.back()=='1') {
54             cout<<s<<' '<<diff<<'
'<<weekday[diff%7]<<'\n';
55             cnt++;
56         }
57
58         s.pop_back();
59         s.pop_back();
60
61     }
62     s.pop_back();
63     s.pop_back();
64 }
65 for(int i=1;i<=4;i++) {
66     s.pop_back();
67 }
68 Months[2]=28; // 置回来
69 s.clear();
70 }
71 cout<<cnt;
72 return 0;
73 }
74
75 /*
76     输入样例:
77
78     输出样例:
79
80 */

```

第五题

- 应该有单调性规律，但是没有推，硬枚举的，填空题嘛

```
1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4  #define int long long
5
6  using namespace std;
7
8  typedef long long LL;
9  typedef pair<int,int> PII;
10
11  /*
12   解题思路：贪心思想，
13
14   */
15
16  const int N=1e5+5;
17  const int Num[]={0, 9226, 4690, 4873, 1285, 4624,
18   1596, 6982, 590,
19   8806, 121, 8399, 8526, 5426, 64, 9655, 7705,
20   3929, 3588, 7397,
21   8020, 1311, 5676, 3469, 2325, 1226, 8203,
22   9524, 3648, 5278, 8647};
23
24  int n,a[N];
25
26  signed main() {
27  //  cout<<(3^5)<<endl; // 测试一下
28  //  int ans=INT_MAX; // 找最小平方和
29  //  int sum=0;
30  //  int idx;
31  //  枚举一个数嘛，就硬枚举，单调性不推了，从1~1e6
32  for(int i=1;i<=1e7;i++) {
33      sum=0;
34      for(int j=1;j<=30;j++) {
35          sum+= (i^Num[j])*(i^Num[j]);
36      }
37  }
```



```

34         if(sum<ans) {
35             ans=sum;
36             idx=i;
37             cout<<idx<<' ' <<sum<<'\n';
38         }
39     }
40     cout<<idx<<' ' <<ans;
41     return 0;
42 }
43
44 /*
45     输入样例:
46
47     输出样例:
48
49 */

```

第六题

- 不足15不计，考模运算

```

1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4  #define int long long
5
6  using namespace std;
7
8  typedef long long LL;
9  typedef pair<int,int> PII;
10
11 /*
12     解题思路:
13

```

```

14  */
15
16  const int N=1e4+5;
17
18  int n,a[N];
19
20  signed main() {
21      cin>>n;
22      cout<<2*floor(n/15);
23
24      return 0;
25  }
26
27  /*
28      输入样例:
29
30      输出样例:
31
32  */

```

第七题

- 数字很大，拆位很麻烦，要开longlong，直接字符串找最大字符

```

1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4
5  using namespace std;
6
7  typedef long long LL;
8  typedef pair<int,int> PII;
9
10 /*

```

```

11      解题思路：每个非零位都-1，操作次数无非就是看整数里面最
      高位是多少
12      10^10 用字符串来存即可
13
14      */
15
16      const int N=1e5+5;
17
18      string s;
19
20      int main() {
21          cin>>s;
22          char mmax=CHAR_MIN;
23          for(int i=0;i<s.size();i++) {
24              if(s[i]>mmax) {
25                  mmax=s[i];
26              }
27          }
28          cout<<mmax;
29          return 0;
30      }
31
32      /*
33      输入样例：
34
35      输出样例：
36
37      */

```

第八题

- scanf格式化输入秒了，注意开longlong

```
1 #include<bits/stdc++.h>
```

```

2 #define x first
3 #define y second
4 #define int long long
5
6 using namespace std;
7
8 typedef long long LL;
9 typedef pair<int,int> PII;
10
11 /*
12     解题思路： 不保证结果非负是吧， 如果是负数多输出一个 '-' 即可
13
14     */
15
16 const int N=1e5+5;
17
18 int a,b;
19
20 signed main() {
21     scanf("%lld-%lld",&a,&b);
22     cout<<a-b;
23
24     return 0;
25 }
26
27 /*
28     输入样例：
29
30     输出样例：
31
32     */

```

第九题

- 区间DP青春版，注意每个数最大 10^5 ，数最多 10^6 ，应开longlong

```
1 #include<bits/stdc++.h>
2 #define x first
3 #define y second
4 #define int long long
5
6 using namespace std;
7
8 typedef long long LL;
9 typedef pair<int,int> PII;
10
11 /*
12     解题思路：间隔为1的k个数之和最大
13     你的意思是可能有 $10^6$ 个大小为 $10^5$ 个数相加！天呐！long
14     long
15     */
16
17 const int N=1e5+5;
18
19 int n,a[N],k;
20
21 signed main() {
22     cin>>n;
23     for(int i=1;i<=n;i++) {
24         scanf("%lld",&a[i]);
25     }
26     cin>>k;
27     int ans=LONG_LONG_MIN;
28
29     // 遍历数组
30     // 遍历起点，且限制边界
31     for(int i=1;i+2*k<=n;i++) {
32         int tmp=0;
33         // 遍历次数，i, i+1*k, i+2*k
```

```

34         for(int j=0;j<k;j++) {
35             tmp+=a[i+j*k];
36         }
37         // 打擂
38         if(tmp>ans) {
39             ans=tmp;
40         }
41     }
42     cout<<ans;
43     return 0;
44 }
45
46 /*
47     输入样例:
48
49     输出样例:
50
51 */

```

第十题

- 打勾序列无非是先递减再递增，找最大序列长度，所以是DP问题
- 对每个元素求在其之前的最大递减子序列长度和最大递增子序列长度，计算元素个数用 $r[i]-l[i]+1$ 即可

```

1  #include<bits/stdc++.h>
2  #define x first
3  #define y second
4
5  using namespace std;
6
7  typedef long long LL;
8  typedef pair<int,int> PII;

```

```

9
10  /*
11     解题思路： 很显而易见的动态规划， 要先递增再递减， 长度最
12     长
13     最开始还想长度最长是不是要用二分来找
14     定义l和r数组， 存储以每个元素结尾的
15     最长递减子序列长度 和 最长递增子序列长度
16
17     */
18  const int N=1e5+5;
19
20  int n,a[N],l[N],r[N];
21
22  int main() {
23
24      cin>>n;
25      for(int i=1;i<=n;i++) {
26          scanf("%d",&a[i]);
27      }
28
29      // 初始化一下
30      for(int i=1;i<=n;i++) l[i]=r[i]=1;
31
32      // 注意方向
33      // DP 找left数组， 对每个元素找前面有多少个元素比它大
34      (递减)
35      for(int i=2;i<=n;i++) {
36          for(int j=1;j<i;j++) {
37              // a[i]<a[j] 说明j可以拼在i的后面(最长递
38              减子序列)
39              if(a[j]>a[i]) l[i]=max(l[i],l[j]+1);
40          }
41      }
42      // for(int i=1;i<=n;i++) cout<<l[i]<<' ';
43      // puts("");

```

```

42
43     // DP 找right数组，对每个元素找前面有多少个元素比它
    小(递增)
44     for(int i=n-1;i>=1;i--) {
45         for(int j=n;j>i;j--) {
46             if(a[j]>a[i]) r[i]=max(r[i],r[j]+1);
    // 拼在后面
47         }
48     }
49     // for(int i=1;i<=n;i++) cout<<r[i]<<' ';
50     // puts("");
51
52     // 最后长度为多少，即是看这个元素前面多少个，后面多少个
53     int ans=INT_MIN;
54     for(int i=1;i<=n;i++) {
55         if(l[i]+r[i]-1>=ans) ans=l[i]+r[i]-1;
56     }
57     cout<<ans;
58     return 0;
59 }
60
61 /*
62     输入样例：
63     10
64     2 1 4 7 4 8 3 6 4 7
65     输出样例：
66     5
67     */

```