

2023 年山东省信息学小学组 (CSP-X) 第一轮试题

考试时间: 2023 年 9 月 16 日 14:30-16:30

考生注意事项:

(1) 试题纸共有 12 页, 答题纸共有 1 页, 满分 100 分。请在答题纸上作答, 写在试题纸上的一律无效。

(2) 不得使用任何电子设备 (如计算器、手机、电子词典等) 或查阅任何书籍资料。

一、单项选择题(共 15 题, 每题 2 分, 共计 30 分; 每题有且仅有一个正确选项)

1. CSP-J/S: CCF 非专业级软件能力认证(Certified Software Professional Junior/Senior, 简称 CSP-J/S)创办于 () 年。

- A. 2018 B. 2019 C. 2020 D. 2021

2. 十进制数 2023 等值于二进制数 ()。

- A. 11100000111 B. 11000001111 C. 10100010111 D. 11111100111

3. 中缀表达式 $F-(B+C/D)*E$ 的后缀形式是 ()。

- A. $FB-C+D/E^*$ B. $FBC+D/-E^*$ C. $FBCD/E^*+-$ D. $FBCD/+E^*-$

4. 对二叉排序树进行 () 遍历, 可以得到该二叉树所有结点构成的有序序列。

- A. 前序 B. 中序 C. 后序 D. 按层次

5. 递归过程或函数调用时, 处理参数和返回地址, 通常使用一种称为 () 的数据结构

- A. 栈 B. 队列 C. 树 D. 链表

6. 某有序数列有 1000 个各不相同的元素, 由小到大排列, 现要对该数列进行二分查找, 在最坏的情况下, 需要查找 () 个元素。

- A. 1000 B. 500 C. 10 D. 8

7. 将两个各有 100 个元素的有序表合并成一个有序表, 最坏情况下的比较次数是 ()。

- A. 100 B. 199 C. 200 D. 99

8. 线性链表不具有的特点是 ()。

- A. 随机访问
B. 不必事先估计所需存储空间大小
C. 插入与删除时不必移动元素
D. 所需空间与线性表长度成正比

9. 二叉树的先序遍历和中序遍历如下:

先序遍历: EFHIGJK 中序遍历: HFIEJKG

该二叉树根的左子树的根是 ()。

- A. E B. F C. G D. H

10. 下面关于算法的描述不正确的是 ()。

- A. 算法必须有输出
B. 算法必须在计算机上用某种语言实现
C. 算法不一定有输入
D. 算法必须在有限步执行后能结束

11. 下列代码执行后, s 的值是 ()。

```
int s=0;
for(int i=1;i<=100;i++)
    for(int j=1;j<=i;j++) s++;
cout<<s<<endl;
```

- A. 100 B. 10000 C. 200 D. 5050

12. 若已知一个栈的入栈顺序是 1, 2, 3, ..., 100, 其输出序列为 $P_1, P_2, P_3, \dots, P_{100}$,

若 P_1 是 100, 则 P_i 是 ()。

- A. i B. 100-i C. 101-i D. 不确定

13. 对于给定的数字 11223344, 如果每次取连续的一段(长度至少为 1)构成一个新的数字, 问能构成多少个不同的数字 ()。

- A. 14 B. 27 C. 32 D. 36

14. 以下函数的时间复杂度为 ()。

```
int f(int n){
    if(n==0||n==1) return 1;
    return f(n/2)+f(n/2)+f(n/2)+f(n/2);
}
```

- A. $O(n \log n)$ B. $O(n^2)$ C. $O(n^2 \log n)$ D. $O(2^n)$

15. 从 6 个白球和 3 个黑球中选出 4 个球, 至少有一个黑球的方案数是 ()。

A. 111

B. 2664

C. 60

D. 360

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 2 分，选择题 3 分，共计 40 分）

(1)

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 int n,ans;
04
05 int main(){
06     cin>>n;
07     for(int i=1;i<=n;i++){
08         for(int j=(i<<1);j<=n;j+=i){
09             ans+=(j^i)==j-i;
10         }
11     }
12     cout<<ans;
13     return 0;
14 }
```

保证输入的 n 不超过 10^6 ，完成下面的判断题和单选题。

● 判断题

16. (1 分) 删除第 2 行，程序能正常运行。()

17. 将第 8 行 $j=(i<<1)$ 改成 $j=2*i$ ，程序输出结果不会发生改变。()

18. 当输入为“3”时，输出为“1”。()

19. 当输入为“6”时，输出为“4”。()

● 单选题

20. 输入“10”时，程序的输出为 ()。

A. 9 B. 8 C. 7 D. 6

21. 程序的时间复杂度为 ()。

A. $O(n)$ B. $O(n\log n)$ C. $O(n^2)$ D. 不确定

(2)

```
01 #include<iostream>
02 using namespace std;
03 int n,top;
04 int a[1000],stack[1000],ans[1000];
05
```

```

06 int main(){
07     cin>>n;
08     for(int i=1;i<=n;i++){
09         cin>>a[i];
10     }
11     for(int i=1;i<=n;i++){
12         while(top && a[i]>a[stack[top]]){
13             ans[stack[top]] = i;
14             top--;
15         }
16         stack[++top]=i;
17     }
18     for(int i=1;i<=n;i++){
19         cout<<ans[i]<<" ";
20     }
21     return 0;
22 }

```

完成下面的判断题和单选题:

● 判断题

22. (1 分) 第 1 行把头文件 `#include<iostream>` 改成万能头文件 `#include<bits/stdc++.h>` 程序能正常运行。 ()
23. 将第 12 行 “>” 改成 “>=”, 程序输出结果可能发生改变。 ()
24. 如果第一行输入 2000, 第二行输入 2000 个以空格隔开的整数 1, 程序输出结果全为 0。 ()
25. 如果输入的数组 `a` 单调递减, 则程序输出结果全为 0。 ()

● 单选题

26. 针对下列输入数据, 程序的输出为 ()。

6 3 3 4 1 5 2

- A. 2 3 5 5 0 0 B. 2 3 5 5 0 5 C. 3 3 5 5 0 0 D. 3 3 5 5 0 5

27. 程序的时间复杂度为 ()。

- A. $O(n)$ B. $O(n \log n)$ C. $O(n^2)$ D. 不确定

(3)

```

01 #include<bits/stdc++.h>
02 using namespace std;
03 const int N=1e6+5;

```



```

04 int n,k;
05 int nums[N];
06 int partition(int left,int right){
07     int pivot=nums[right];
08     int i=left-1;
09     for(int j=left;j<right;j++){
10         if(nums[j]<=pivot){
11             i++;
12             swap(nums[i],nums[j]);
13         }
14     }
15     swap(nums[i+1],nums[right]);
16     return i+1;
17 }
18
19 int quickSelect(int left, int right, int k) {
20     if (left == right) {
21         return nums[left];
22     }
23     int pivotIndex = partition(left, right);
24     if (k == pivotIndex) {
25         return nums[k];
26     } else if (k < pivotIndex) {
27         return quickSelect(left, pivotIndex - 1, k);
28     } else {
29         return quickSelect(pivotIndex + 1, right, k);
30     }
31 }
32
33 int main() {
34     cin>>n>>k;
35     for(int i=1;i<=n;i++) cin>>nums[i];
36     int ans=quickSelect(1, n, k);
37     cout<<ans<<endl;
38     return 0;
39 }

```

保证输入的 n 不超过 10^6 , k 不超过 n , 且 $1 \leq a_i \leq 10^9$ 。完成下面的判断题和单选题。

● 判断题

28. (1 分) 上述代码正确运行后, 可以将 num 数组按从小到大的顺序排序。 ()

29. 如果将程序第 10 行 " \leq " 改成 " $<$ ", 程序运行结果不变。 ()

30. 该程序实现的功能是求 n 个数中第 k 大的数。 ()

● 单选题

31. 对于以下输入数据, 输出结果为 ()。

10 6

8 3 4 100 2 1 23 45 8 1 50

- A. 4 B. 8 C. 23 D. 45

32. 对于以下输入数据:

10 4
1 2 6 4 3 10 9 7 8 5

程序运行结束时， num 数组内的值为 ()。

- A. [1,2,3,4,5,6,7,8,9,10] B. [1,2,3,4,5,10,9,8,7,6]
C. [1,2,3,4,5,10,9,7,8,6] D. [3,2,1,4,5,10,9,8,7,6]

33. 该程序时间复杂度为 ()。

- A. $O(\log n)$ B. $O(n)$ C. $O(n \log n)$ D. $O(n^2)$

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

(1) 高精度减法

输入两个高精度数 a 和 b, 输出 a-b 的值。

程序中使用了运算符重载: 运算符重载, 就是对已有的运算符重新进行定义, 赋予其另一种功能, 以适应不同的数据类型的运算。

```
#include<cstdio>
#include<iostream>
#include<cstring>
using namespace std;
const int N=10100;
struct big{
    int d[N];
    int len;
};
big a,b;
char s[N];
```

```

big read(){//读入大整数

    big c;
    memset(c.d,0,sizeof(c.d));
    scanf("%s",s+1);
    int n=strlen(s+1);
    for(int i=1;i<=n;i++)c.d[i]= ① ;
    c.len=n;
    return c;
}

bool operator >=(big a,big b){//重载>=
    int n=a.len,m=b.len;
    if(n>m)return 1;
    if(n<m)return 0;
    for(int i=n;i>0;i--){
        if(a.d[i]>b.d[i])return 1;
        if(a.d[i]<b.d[i])return 0;
    }
    return 1;
}

big operator -(big a,big b){//重载-
    int n=a.len;
    for(int i=1;i<=n;i++){
        if( ② ){
            a.d[i+1]--;
            ③ ;
        }
        a.d[i]-=b.d[i];
    }
}

```

```

        while( ④ )n--;
        a.len=n;
        return a;
    }
void write(big a){//输出大整数
    for(int i=a.len;i>0;i--)printf("%d",a.d[i]);
    printf("\n");
}
int main(){
    a=read();
    b=read();
    if(a>=b){
        write(a-b);
    }
    else{
        printf("-");
        ⑤ ;
    }
    return 0;
}

```

单项选择

34. ①处应填 ()

- A. s[n-i] B. s[n-i]-'0' C. s[n+1-i] D. s[n+1-i]-'0'

35. ②处应填 ()

- A. a.d[i]>b.d[i] B. a.d[i]<b.d[i]
C. a.d[i]=b.d[i] D. a.d[i]==b.d[i]

36. ③处应填 ()

- A. a.d[i]=10 B. a.d[i]=+10 C. a.d[i]-=10 D. a.d[i]+=10

37. ④处应填 ()

- A. `a.d[n]==0` B. `n>1` C. `a.d[n]==0&& n>1` D. `a.d[n]==0|| n>1`

38. ⑤处应填 ()

- A. `write(b-a)` B. `b-a` C. `write(a-b)` D. `-write(a-b)`

(2) 求区间最值

给定序列 a_n ，你需要回答 q 次询问，每次询问一个区间 $[l,r]$ 内的最大值与最小值之差。

数据范围满足 $n, q \leq 100000, 1 \leq l \leq r \leq n, a_i \leq 1000000$ 。

提示：每次询问暴力去求区间最值很显然超时，因此我们采用分块算法，分块算法如下：

1、分块：将序列分成等长的 \sqrt{n} 块，其中每块长度也为 \sqrt{n} ，预处理并记录每个元素所属的块以及每块的左右端点下标、最大值和最小值。

2、查询：如果查询区间在同一块内，则暴力扫描统计区间最大最小值；否则，如果查询区间包含多个块，统计除去头尾两个块的中间每个块的已经维护好的最大最小值，然后再暴力统计左端点所在块以及右端点所在块的最大最小值。

```
#include<bits/stdc++.h>
using namespace std;
const int N=2e5+5;
int n,q,block,num;
int a[N],L[N],R[N],belong[N],block_max[N],block_min[N];
void build(){
    block=sqrt(n*1.0);
    num=n/block;
    if(n%num) num++;
    for(int i=1;i<=num;i++){ //每块左右端点下标
        ____①____;
    }
    R[num]=n;
```

```

for(int i=1;i<=n;i++) //每个元素所属块号
    belong[i]=__②__;
for(int i=1;i<=num;i++){
    int minn=1e9,maxx=-1e9;
    for(int j=L[i];j<=R[i];j++){
        maxx=max(maxx,a[j]);
        minn=min(minn,a[j]);
    }
    block_max[i]=maxx;
    block_min[i]=minn;
}
}

int query(int l,int r){
    int minn=1e9,maxx=-1e9;
    if(__③__){
        for(int i=l;i<=r;i++){
            maxx=max(maxx,a[i]);
            minn=min(minn,a[i]);
        }
        return maxx-minn;
    }else{
        for(int i=l;i<=R[belong[l]];i++){
            maxx=max(maxx,a[i]);
            minn=min(minn,a[i]);
        }
        for(__④__){
            maxx=max(maxx,block_max[i]);

```

```

        minn=min(minn,block_min[i]);
    }
    for(int i=L[belong[r]];i<=r;i++){
        maxx=max(maxx,a[i]);
        minn=min(minn,a[i]);
    }
    return maxx-minn;
}
}

int main() {
    cin>>n>>q;
    for(int i=1;i<=n;i++) cin>>a[i];
    build();
    while(q--){
        int l,r;
        cin>>l>>r;
        cout<<___⑤___<<endl;
    }
    return 0;
}

```

单选题

39. ①处应填（ ）。

- A. $L[i]=(i-1)*block, R[i]=i*block$
- B. $L[i]=(i-1)*block+1, R[i]=i*block$
- C. $L[i]=i*block+1, R[i]=(i+1)*block$
- D. $L[i]=i*block, R[i]=(i+1)*block$

40. ②处应填 ()。

- A. $(i-1)/\text{block}+1$ B. $(i-1)/\text{block}$ C. $i/\text{block}+1$ D. i/block

41. ③处应填 ()。

- A. $r-l+1<\text{block}$ B. $\text{belong}[l]==\text{belong}[r]$
C. $l/\text{block}==r/\text{block}$ D. $l/\text{block}+1==r/\text{block}$

42. ④处应填 ()。

- A. `int i=belong[l]; i<=belong[r]; i++`
B. `int i=R[belong[l]]; i<L[belong[r]]; i++`
C. `int i=l/block+1; i<=r/block+1; i++`
D. `int i=belong[l]+1; i<=belong[r]-1; i++`

43. ⑤处应填 ()。

- A. `query(l-1,r)` B. `query(l,r)`
C. `query(l+1,r)` D. `query(l+1,r-1)`

2023 年山东省信息学小学组（CSP-X）第一轮试题答案

一、单项选择题（共 15 题，每题 2 分，共计 30 分）

1	2	3	4	5	6	7	8	9	10
B	D	D	B	A	C	B	A	B	B
11	12	13	14	15					
D	C	C	B	A					

二、阅读程序（判断题 2 分，选择题 3 分，共计 40 分）

第一题	判断题(填√或×)				单选题	
	16（1 分）	17	18	19	20	21
	×	√	√	×	D	B
第二题	判断题(填√或×)				单选题	
	22（1 分）	23	24	25	26	27
	×	√	×	√	C	A
第三题	判断题(填√或×)			单选题		
	28（1 分）	29	30	31	32	33
	×	√	×	B	C	B

三、完善程序（单选题，每小题 3 分，共计 30 分）

第一题					第二题				
34	35	36	37	38	39	40	41	42	43
D	B	D	C	A	B	A	B	D	B