

---

一. 单项选择题 (共20题, 每题1.5分, 共计30分。每题有且仅有一个正确答案。)

1.  $(2019)_{12} + (9102)_{16} =$ :

A:  $(1001100110100111)_2$  B:  $(116643)_8$  C:  $(9DA7)_{16}$  D:  $(9DA5)_{16}$

2. 图灵奖是信息学的最高奖项, 以下获得过图灵奖的中国人是:

A: 姚期智 B: 姚期辉 C: 马云 D: 马化腾

3. 国际信息学奥林匹克竞赛缩写是:

A: NOI B: CTSC C: IOI D: ACM

4.  $2.0E-3 =$

A: 2000 B: 0.002 C: 8 D: -2000

5. 计算  $2019 >> 6 \& 1 =$

A: 1 B: 31 C: 0 D: 2019

6. 使用二分算法在一个大小为  $n$  ( $n \geq 4$ ) 中寻找第4大的整数所需的时间复杂度为:

A:  $O(1)$  B:  $O(n \log n)$  C:  $O(\log n)$  D:  $O(n)$

7. 若设函数  $f(x) = \begin{cases} 1 & (x=1, x=2) \end{cases}$

$3 * f(\sqrt{x}) + f(x/2) + 1 \quad (x > 2)$

当  $x=19$  时, 计算过程中共调用的  $f(x)$  个数是 (包括调用  $f(1)$ ,  $f(2)$ ):

注释: 此处运算默认下取整

A: 3 B: 4 C: 5 D: 6

8. 上题函数中  $f(19) =$

A: 30 B: 37 C: 36 D: 39

9. 第7题中的函数值**不可以**用以下哪种方法求得:

A: 动态规划 B: 分治 C: 递推 D: 递归搜索

10.以下部件损坏，主机仍可正常工作的是：

A: 内存条      B: 硬盘      C: 显示屏      D: 显卡

11.对一下数据{1000, 2,3,5,4,1, 5000}进行冒泡排序，共计需交换次数为：

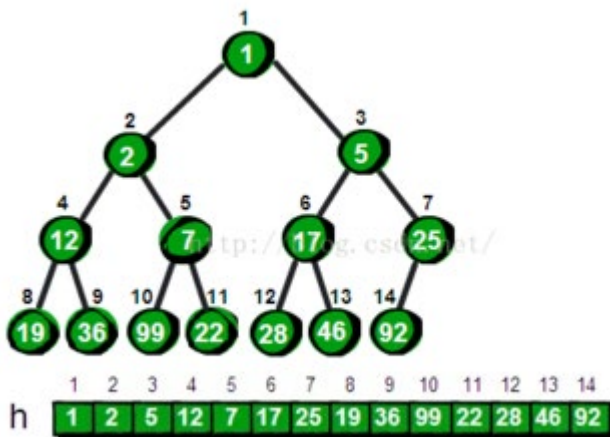
A: 5      B: 10      C: 15      D: 18

12.如果将人体比作计算机，那么人体的记忆中枢相当于以下计算机部件的：

A: 运算器      B: 中央处理器      C: 控制器      D: 内存

13.以下示意图中的数据结构**不属于**选项中的哪个数据结构：

A: 大根堆      B: 无向图      C: 连通图      D: 完全二叉树



14.dos、unix和windows的共同点是：

A: 都是硬件      B: 都是联网系统软件      C: 都是应用软件      D: 都已

经过时

15.html是一种高级语言，以下操作可以查看html代码的是：

A: 打开浏览器按F11      B: 运行html.exe

C: 无法查看      D: 打开浏览器按F12

16.以下关于计算机病毒的说法正确的是：

---

A: 防火墙可以防止感染

B: 通过生物传播

C: 一旦感染无法破解

D: 计算机一次感染终身免疫

17.c++语言“实数下取整”操作是:

A: (int)x

B: float(x)

C: floor(x)

D: ceil(x)

18.一棵n层二叉树的最多节点数减去最少节点数等于:

A:  $2^n$

B:  $2^n - n$

C:  $n^2 - n$

D:  $n \cdot \log_2(n) - n$

19.现给出以下程序:

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

```
using namespace std;
```

```
int i,x;
```

```
int a[11]={0,10,2,3,5,14,8,20,1,7,-1};
```

```
int main()
```

```
{
```

```
    cin>>x;
```

```
    sort(a+1,a+11);
```

```
    for (i=1;i<=10;i++)
```

```
        if (a[i]>=x) break;
```

```
        cout<<i<<endl;
```

```
}
```

问若将此程序的输入输出看做函数，则此函数的图像不经过点:

A: (0,2)

B: (2,4)

C: (11,9)

D: (21,11)

20.上题程序划线部分可替换为:

A: `cout<<upper_bound(a,a+10,x)<<endl;`

B: `cout<<upper_bound(a+1,a+11,x)<<endl;`

C: `cout<<upper_bound(a+1,a+11,x)-a<<endl;`

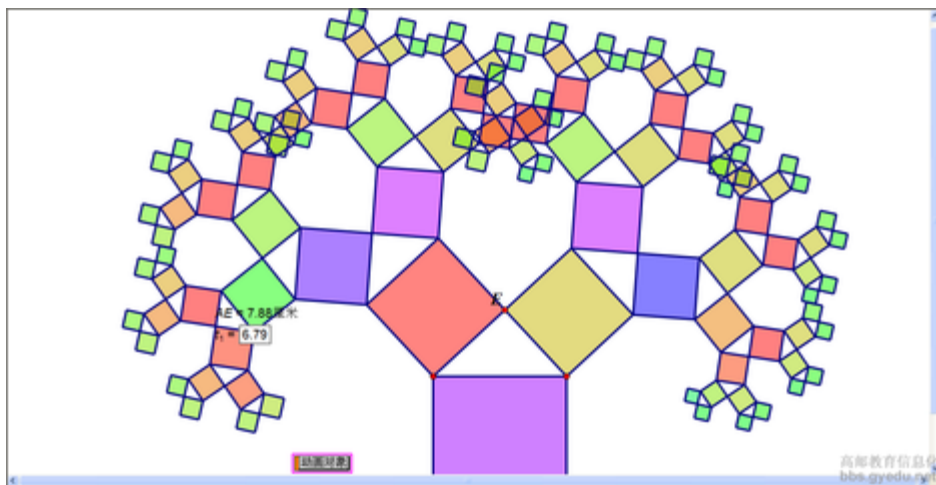
D: `cout<<lower_bound(a+1,a+11,x)-a<<endl;`

## 二. 问题求解（共2题，每题5分，共计10分）

1.五位数的卡布列克运算循环节为：\_\_\_\_\_

注释：卡布列克运算为将一数的所有数位数字重新排列可得的数的最大值减最小值（高位补零），保证有循环节，本题有三个答案，写出一个即得5分，各数字用逗号隔开。

2.对于一棵勾股树（任一直角三角形三边均有与边长等长正方形重合，任一直角三角形直角边为边长的正方形均与另一直角三角形斜边重合，如图），设所有最小正方形边长为 $a[i] (1 \leq i \leq \infty)$ ，则最大正方形面积为\_\_\_\_\_ ( $1 \leq i \leq \infty$ )



## 三. 阅读程序写结果（共4题，每题8分，共计32分）

1.

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

---

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,b,c;
```

```
    double ans;
```

```
    cin>>a>>c;
```

```
    if (!c>>1<<1==c) c-=1;
```

```
    b=(a*c)/2;
```

```
    ans=sqrt(pow(b,3));
```

```
    printf("%.2f",ans);
```

```
    return 0;
```

```
}
```

输入： 1 3

输出： \_\_\_\_\_

2.

```
#include<iostream>
```

```
#include<algorithm>
```

```
using namespace std;
```

```
int n,a[101],i;
```

```
int main()
```

```
{
```

```
    cin>>n;
```

---

```

    for (i=1;i<=n;i++) cin>>a[i];

    sort(a+1,a+n+1);

    n=unique(a+1,a+n+1)-a-1;

    cout<<n<<endl;

    for (i=1;i<=n;i++) cout<<a[i]<<' ';

    cout<<endl;

    return 0;

}

```

输入： 10

20 40 32 67 40 20 89 300 400 15

输出： \_\_\_\_\_

3.

```

#include<bits/stdc++.h>

using namespace std;

long long o=1,minn=10000000,m;

struct pa{

    long long s;

    long long j;

    string n;

    long long cost;

};

pa  p[10005];

```

---

```
int main()
{
    while(cin>>p[o].s>>p[o].j>>p[o].n)
        o++;
    for(int i=1;i<o;i++)
    {
        for(int g=1;g<o;g++)
            p[i].cost+=abs(p[i].j-p[g].j)*p[g].s;
        if(p[i].cost<=minn)
        {
            minn=p[i].cost;
            m=i;
        }
    }
    cout<<p[m].n<<" "<<p[m].cost<<endl;
    return 0;
}
```

输入： 7 9289 Vladivostok

5 8523 Chabarovsk

3 5184 Irkutsk

8 2213 Yalutorovsk

10 0 Moscow

---

输出: \_\_\_\_\_

4.

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

```
using namespace std;
```

```
int a[500001],b[500001],i,n,A,B,l,r,mid;
```

```
bool check(int mid)
```

```
{
```

```
    int ii,s=0;
```

```
    memcpy(b,a,sizeof(a));
```

```
    for (ii=1;ii<=n;ii++) b[ii]-=mid*A;
```

```
    for (ii=1;ii<=n;ii++)
```

```
        if (b[ii]>0) s+=(int)ceil((double)b[ii]/B);
```

```
    return s<=mid;
```

```
}
```

```
int main()
```

```
{
```

```
    scanf("%d%d%d",&n,&A,&B);
```

```
    for (i=1;i<=n;i++) scanf("%d",&a[i]);
```

```
    l=0;
```

```
    r=500010;
```

```
    while (l<r)
```

```
{
```



---

```
        mid=(l+r)/2;

        if (check(mid)) r=mid;else l=mid+1;

    }

    printf("%d\n",l);

}
```

输入：

10 2 3

9 10 3 12 7 4 8 15 9 24

输出：\_\_\_\_\_

四. 完善程序 (前8空, 每空2分, 后4空, 每空3分, 共28分)

1. **逆序对**：逆序对的定义是：对于给定的一段正整数序列，逆序对就是序列中 $a_i > a_j$ 且 $i < j$ 的有序对。逆序对可以用冒泡排序和归并排序求得。试完善以下冒泡排序程序与归并排序程序。

```
s=0;

for (i=1;i<=n;i++)

    for (j=1;j<=____(1)____;j++)

        if (a[j]>a[j+1])

        {

            _____(2)_____;

            ++s;

        }

cout<<s<<endl;
```

---

////////////////////////////////////

#include<bits/stdc++.h>

using namespace std;

long long a[500001],b[500001],s,n;

void guibing(long long l,long long r)

{

if (r-l==0) return;

if (r-l==1)

{

if (a[r]<a[l])

{

swap(a[r],a[l]);

++s;

}

return;

}

long long mid= \_\_\_\_ (3) \_\_\_\_,i=l,j=mid+1,k=l;

guibing(l,mid);\_\_\_\_ (4) \_\_\_\_;

while (\_\_\_\_ (5) \_\_\_\_)

if (a[i]>a[j])

{

s+= \_\_\_\_ (6) \_\_\_\_

---

```
        b[k]=a[j];

        ++k;

        ++j;

    }

    else

    {

        b[k]=a[i];

        ++k;

        ++i;

    }

    for (;i<=mid;i++) b[k++]=a[i]; for (;j<=r;j++) b[k++]=a[j];

    for (k=l;k<=r;k++)       (7)      ;

}

int main()

{

    long long i;

    scanf("%ld",&n);

    for (i=1;i<=n;i++) scanf("%ld",&a[i]);

    guibing(1,n);

    printf("%ld\n",s);

    return 0;

}
```

---

**2.石子合并：** 在一个圆形操场的四周摆放 N 堆石子,现要将石子有次序地合并成一堆.规定每次只能选相邻的 2 堆合并成新的一堆，并将新的一堆的石子数，记为该次合并的得分。试设计出 1 个算法,计算出将 N 堆石子合并成 1 堆的最小得分和最大得分.

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

using namespace std;

const int Maxn=1000+10;

int t[Maxn],dp1[Maxn][Maxn],dp2[Maxn][Maxn];

int main(){

    int n;

    cin>>n;

    for(int i=1;i<=n;++i) {

        cin>>t[i];

        t[___(1)___]=t[i];}

    for(int i=2;i<=2*n;++i) t[i]+=t[i-1];

    memset(dp1,0,sizeof(dp1));

    for(int i=2*n-1;    (2)    )

        for(int j=    (3)    ;j<=2*n;++j)

            {

                int b=2147483647;

                for(int k=i;k<j;++k){

                    dp1[i][j]=max(dp1[i][j],dp1[i][k]+dp1[k+1][j]+t[j]-t[i-1]);

                    b=min(b,___(4)___);}

                dp2[i][j]=b; }
```

---

```
int max0=0,min0=100000000;

for(int i=1;i<=n;++i){

    max0=max(____(5)____);

    min0=min(min0,dp2[i][i+n-1]);}

cout<<min0<<endl<<max0<<endl;

return 0;

}
```