一、选择题（每题2分，共计30分，每题有4个选项，只有一个答案正确）

1、在下列几种存储器中，访问速度最慢的是（）。A

A、ROM B、cache C、RAM D、寄存器

2、在等腰直角三角形ABC中，过直角顶点C在∠ACB内部任作一射线CM，与线段AB交于点M，求AM<AC的概率。 （）D

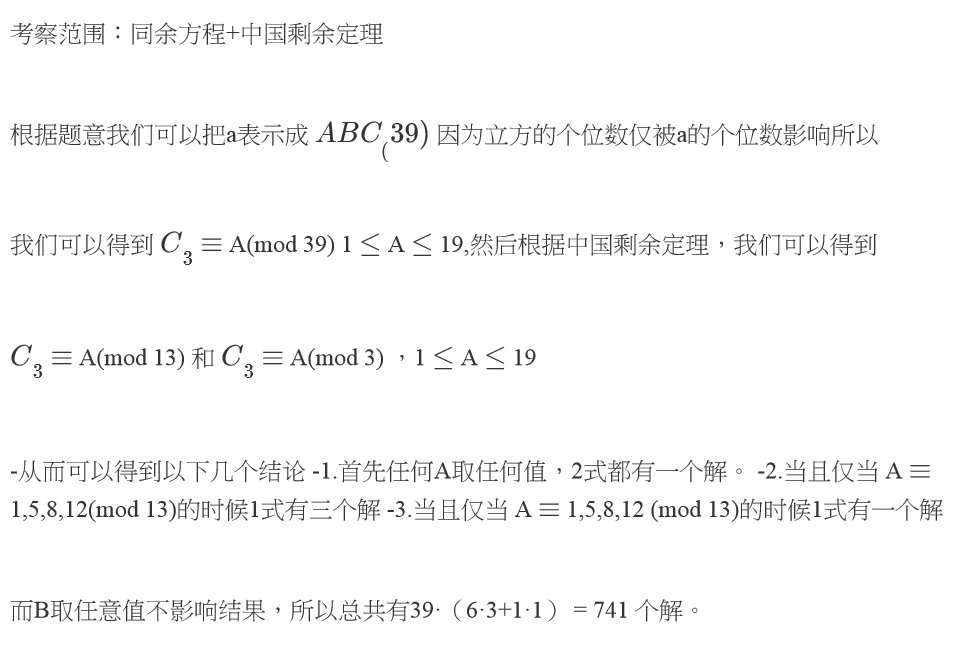
A、1/sqrt(2) B、1/2 C、sqrt(2)/2 D、3/4

解析：画图。注意是看弧而不是看AB上的比例，弧度是67.5度/90度=3/4

3、有一个整数a 满足 392 < a < 20·392 满足以下条件 当a和 a3 书写在39进制下时，a3 的末尾和a的最高位相同。 求有多少个满足条件的a。-by chr D

A.39 B.429 C.702 D.741

解析：



4、若某算法的计算时间表示为递推关系式：

T(N) = 2T(N / 4) + N^(1/2)

T(1) = 1

则该算法的时间复杂度为（ ）。 C

A. O(N^(1/2)) B. O(N log N) C. O(N^(1/2)logN) D. O(N^2)

5、以下程序的输出结果是（ B ）

int main(){

char st[20]= "hello\0\t\\\";

printf(“%d %d \n",strlen(st),sizeof(st));

}

A. 12 20 B. 5 20 C. 20 20 D. 5 5

6、设某强连通图中有n个顶点，则该强连通图中至少有（ ）条边。C

A. n(n-1) B. n+1 C. n D. n(n+1)

7、如果在某个进制下等式7\*7=41成立，那么在该进制下12\*12=( B )也成立。

A. 100 B. 144 C. 164 D.196

8、关于计算机网络，以下说法正确的是（）C

A.网络就是计算机的集合

B.网络可提供远程用户共享网络资源，但可靠性很差。

C.网络是计算机技术和通信技术相结合的产物。

D.当今世界上规模最大的网络是LAN。

9、若A=False，B=True，C=True，D=False，则下列逻辑运算真的有（）D

A、(A∧B)V(C∧D) B、((A∧¬B)VD)∧B

C、((¬A→¬B)∧C)VD D、¬A∧(¬D→¬C)→¬B

10、在整数的补码表示法中，以下说法正确的是（ ）。A

A．只有负整数的编码最高位为1

B．在编码的位数确定后，所能表示的最小整数和最大整数的绝对值相同

C．整数0有正零和负零两个编码

D．两个用补码表示的数相加时，如果在最高位产生进位，则表示运算溢出

11、在双向链表p结点之前插入结点s的操作，正确的是（）：B

① s->prior=p->prior;

② p->prior->next=s;

③ s->next=p;

④ p->prior=s;

A、④①②③ B、②①③④ C、③①④② D、④③②①

12、下列关于排序的算法，不正确的是（）。B

A.不存在这样一个基于排序码比较的算法：它只通过不超过9次排序码的比较，就可以对任何6个排序码互异的数据对象实现排序。

B.如果输入序列已经排好序，快速排序算法将比归并排序更快。

C.希尔排序的最后一趟就是冒泡排序。

D.任何基于排序码比较的算法，对n个数据对象进行排序时，最坏情况下的时间复杂度不会低于O(nlogn)。

13、下列有关图的说法正确的是（）。C

A. 对于一个边上权值任意的带权有向图，使用Dijkstra算法可以求一个顶点到其它各个顶点的最短路径。

B.一个有向图的拓扑排序，实质上是图上所有顶点按照入度的非降序排列。

C.有回路的有向图不能完成拓扑排序。

D. 对任何用顶点表示活动的网络（AOV网）进行拓扑排序的结果都是唯一的。

14、下列有关说法不正确的是（）。B

A. 区块链的本质是一个去中心化的数据库，它将为人类社会带来巨大的变革。

B. 谷歌人工智能新应用Duplex已经通过了图灵测试。

C. G网络作为第五代移动通信网络，其峰值理论传输速度可达每秒数十GB。

D. 艾德文·卡特姆和帕特·汉拉汉凭借在3D计算机图形学领域的创新获得了2020年的图灵奖。

15、一个团队里面有n个成员，如果有3个人及以上的人互为朋友或者都不是朋友关系，那么我们就说这个团队是bad team。求n的最小值使得这个团队是bad team。（ ） C

A、4 B、5 C、6 D、7

解析：拉姆齐定理

二、阅读程序写结果

1 #include<iostream>

2 using namespace std;

3 int main(){

4 int n;

5 cin>>n;

6 string s="1";

7 int t;

8 do{

9 int i=s.length()-1;

10 while(s[i]=='1'){

11 s[i]='0';i--;

12 }

13 if(i>=0)s[i]='1';

14 else s='1'+s;

15 t=0;

16 for(int i=0;i<s.length();i++){

17 t=t\*10+s[i]-'0';

18 }

19 }while(t%n);

20 cout<< s<<endl;

21 return 0;

22 }

Do while循环里面在枚举二进制数，结果输出能整除n的最小数

16、n=1，答案为1。 F

17、若13行i>=0改成i>0,不影响程序结果。 T

18、程序的时间复杂度为O(n)。 F

19、当输入n为81时，计算结果s数字恰好都为1，请判断是否正确（）？F

20、输入为7，结果为（）？ A

A. 1001 B. 1011 C. 1011 D. 1111

21、输入为41，结果为（）？ D

A. 10001 B. 10011 C. 11011 D. 11111

1 #include<iostream>

2 using namespace std;

3 int f(int n,int k){

4 if (n == k || k == 0) return 1;

5 else return f(n - 1, k) + f(n - 1, k - 1);

6 }

7 int main() {

8 int n,k,m;

9 cin >> n >> k;

10 cout << f(n,k);

11 return 0;

12 }

22、第5行else 可以去掉不影响程序结果。 T

23、第4行’||’换成’&&’不影响程序结果。 F

24、输入（n,k）一定等于(n,n-k+1) F

25、程序的时间复杂度为多项式级别 F

26、输入为6 3，结果为（）？ A

A. 20 B. 24 C. 27 D. 32

27、输入为20 15，结果为（）？ C

A. 4845 B. 12306 C. 15504 D. 38760

#include<stdio.h>

#include<bits/stdc++.h>

#define max 500000

int count[max];

int tree[max][26];

int main() {

int cnt=1,i;

char str[20];

while(gets(str),strlen(str)>0) {

int len=strlen(str);

int flag=0;

for(i=0; i<len; i++) {

if(tree[flag][str[i]-'a']==0) {

tree[flag][str[i]-'a']=cnt;

cnt++;

}

flag=tree[flag][str[i]-'a'];

count[flag]++;

}

}

while(gets(str)) {

int num=0;

int flag1=0;

int len1=strlen(str);

for(i=0; i<len1; i++) {

if(tree[flag1][str[i]-'a']==0) {

break;

}

flag1=tree[flag1][str[i]-'a'];

if(i==len1-1) {

num=count[flag1];

}

}

printf("%d ",num);

}

return 0;

}

统计出以某个字符串为前缀的单词数量

28、第9行’,’改成’&&’ 不影响程序结果。 T

29、第10行“int len=strlen(str);”改为“int len=str.size();”不影响程序结果。 F

30、本程序使用的数据结构是链表。 F

31、如果输入为：

123

312

12

12

结果为3，请判断是否正确（）？F

32、如果输入为：

banana

band

bee

absolute

acm

ba

b

band

abc

结果为（）？B

A.1 0 3 0 B. 2 3 1 0 C. 2 4 1 3 D.2 3 1 4

33、对于上一题输入，flag最大值为（）？ C

A. 5 B. 15 C. 19 D. 31

四、完善程序

问题描述: 给定两个整数 n, m (0 <= m <= n <= 10^18), 求 C(n, m) mod 10^10 的值.

下面的程序使用了扩展卢卡斯算法来解决本题. by-zrf

#include <iostream>

using namespace std;

namespace lucas {

const int maxp = 2e7;

int fact[maxp], p, q;

void Solve(long long n, int &a, int &b) {

long long y = n / q, z = fact[q - 1];

for (; y; y >>= 1) {

if (y & 1) a = (long long)a \* z % q;

z = (long long)z \* z % q;

}

a = (long long)a \* fact[n % q] % q;

if (n >= p) {

b += n / p;

Solve(/\*\_\_\_(1)\_\_\_\*/);

}

}

void Init(int p1, int a1) {

p = p1;

q = 1;

for (int i = 0; i < a1; ++i) {

q \*= p;

}

fact[0] = 1;

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

if (i % p) {

fact[i] = (long long)fact[i - 1] \* i % q;

} else {

fact[i] = fact[i - 1];

}

}

}

void Exgcd(int a, int b, int &d, int &x, int &y) {

if (!b) {

d = a;

x = 1;

y = 0;

} else {

Exgcd(/\*\_\_\_(2)\_\_\_\*/);

y -= /\*\_\_\_(3)\_\_\_\*/;

}

}

int Inv(int a, int m) {

int d, x, y;

Exgcd(a, m, d, x, y);

x %= m;

return x < 0 ? x + m : x;

}

long long Main(long long n, long long m, long long mod) {

int pm[50], cnt[50], top = 0;

long long x = mod;

for (long long i = 2; i \* i <= x; ++i) {

if (x % i == 0) {

pm[top] = i;

cnt[top] = 0;

while (x % i == 0) {

x /= i;

++cnt[top];

}

++top;

}

}

if (x > 1) {

pm[top] = x;

cnt[top++] = 1;

}

int al[50], ar[50];

for (int i = 0; i < top; ++i) {

Init(pm[i], cnt[i]);

int a = 1, b = 0, c = 1, d = 0;

Solve(n, a, b);

Solve(m, c, d);

Solve(n - m, c, d);

a = (long long)a \* Inv(c, q) % q;

for (int j = 0; j < b - d; ++j) {

a = (long long)a \* p % q;

}

al[i] = a;

ar[i] = q;

}

long long ans = 0;

for (int i = 0; i < top; ++i) {

long long tmp = mod / ar[i];

ans += tmp \* Inv(tmp % ar[i], ar[i]) % mod \* al[i] % mod;

if (ans >= mod) /\*\_\_\_(4)\_\_\_\*/;

}

return ans;

}

}

int main(void) {

long long n, m;

cin >> n >> m;

cout << lucas::Main(/\*\_\_\_(5)\_\_\_\*/) << endl;

return 0;

}

DCCBA

34. A. n/p,b,a B. n%p,b,a C. n%p,a,b D. n/p,a,b

35. A. b,a/b,d,x,y B. a,a/b,d,y,x C. b,a%b,d,y,x D. a,a%b,d,x,y

36. A. b/a\*y B. a/b\*y C. a/b\*x D. b/a\*x

37. A. ans/=mod B. ans-=mod C. tmp=ans D. tmp=mod

38. A.n,m,1e10 B. m,n,1e10 C. n%m,m,1e10 D. n/m,m,1e10

2、

题意：给你个序列，你可以给某个数加上2的幂次(非负)，问最少多少次可以让所有的数相等。可以证明在限制条件下答案不会超过1e18.

输入格式：

第一行一个数：n（1<=n<=1e5）

第二行n个数ai (0<=ai<=1e17 )

输出格式：

一个数字

样例输入：

3

2 2 8

样例输出：

3

样例解释：第一个2加上2^3，第二个2加上2^3，第三个数8加上2^1

Code by mangoyang

Codeforces 1188D Make Equal 可以看一下官方题解

#include <bits/stdc++.h>

#define inf (0x7f7f7f7f)

#define Max(a, b) ((a) > (b) ? (a) : (b))

#define Min(a, b) ((a) < (b) ? (a) : (b))

typedef long long ll;

using namespace std;

template <class T>

inline void read(T &x){

int ch = 0, f = 0; x = 0;

for(; !isdigit(ch); ch = getchar()) if(ch == '-') f = 1;

for(; isdigit(ch); ch = getchar()) x = x \* 10 + ch - 48;

if(f) x = -x;

}

const int N = 100005;

ll a[N], b[N];

unsigned long long mod;

int dp[65][N], ans, n;

inline bool cmp(ll x, ll y){ return /\*\_\_\_(1)\_\_\_\*/ ; }

inline void chkmin(int &x, int y){ x = min(x, y); }

int main(){

read(n);

for(int i = 1; i <= n; i++) read(a[i]);

mod = 1ll << 62ll;

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

for(int i = 1; i <= n; i++) b[i] = a[1] - a[i];

memset(dp, 127, sizeof(dp)), dp[0][0] = 0;

for(ll i = 0; i < 64; i++){

mod = 1ll << i;

sort(b + 1, b + n + 1, cmp);

int s0 = 0, s1 = 0, tot = 0, tot1 = 0;

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

s0 +=/\*\_\_\_(2)\_\_\_\*/ ;

s1 += ((1ll << i) & b[j]) ? 0 : 1;

}

tot1 = s0;

for(int j = 0; j <= n; j++) {

if(j > 0){

tot += /\*\_\_\_(3)\_\_\_\*/ ;

s0 += ((1ll << i) & b[j]) ? -1 : 1;

s1 += ((1ll << i) & b[j]) ? 1 : -1;

}

if(dp[i][j] == inf) continue;

chkmin(dp[i+1][tot], dp[i][j] + s0);

chkmin( /\*\_\_\_(4)\_\_\_\*/ , dp[i][j] + s1);

}

}

cout <</\*\_\_\_(5)\_\_\_\*/ << endl;

return 0;

}

39. A. (y % mod) > (x % mod) B. (x % mod) > (y % mod)

C. (y % mod) < (x % mod) D. (x % mod) < (y % mod) B

40. A. ((1ll << i) & b[i]) ? 0 : 1 B. ((1ll << i) & b[i]) ? 1 : 0

C. ((1ll << i) & b[j]) ? 0 : 1 D. ((1ll << i) & b[j]) ? 1 : 0 D

41. A. ((1ll << i) & b[j]) ? 1 : 0 B. ((1ll << j) & b[j]) ? 1 : 0

C. ((1ll << i) & b[j]) ? 0 : 1 D. ((1ll << j) & b[j]) ? 0 : 1 A

42. A. dp[i+1][tot] B. dp[i+1][j+tot-tot1]

C. dp[i+1][j-tot1] D. dp[i+1][j-tot+tot1] D

43. A. dp[0][0] B. dp[64][0]

C. dp[n][0] D. dp[N][0] B