第二十五届全国青少年信息学奥林匹克联赛初赛

CSP-S C++语言模拟试题

竞赛时间:2019 年 10 月 13 日 14:30～16:30

选手注意:

- 试题纸共有 10 页，答题纸共有 2 页，满分 100 分。请在答题纸上作答，写在试题纸上的一律无效。
- 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.以下哪个不是 2019IOI 金牌（）

A.钟子谦

B.杨骏昭

C.王修涵

D.高嘉煊

2.原地排序是指在排序过程中(除了存储待排序元素以外的)辅助空间的大小与数据规模无关的排序算法。以下不属于原地排序的有( )。

A.冒泡排序　　B.插入排序　　C.基数排序　　D.选择排序

3.一个平面的法线是指与该平面垂直的直线。过点(1,1,1)、(0,3,0)、(2,0,0)的平面的法线是( )。

A. 过点(1,1,1)、(2,3,3)的直线

B. 过点(1,1,1)、(3,2,1)的直线

C. 过点(0,3,0)、(-3,1,1)的直线

D. 过点(2,0,0)、(5,2,1)的直线

4．双向链表中有两个指针域 llink 和 rlink，分别指向该结点的前驱及后继。设 p 指向链表中的一个结点，它的左右结点均非空。现要求删除结点 p，则下面语句序列中不正确的是( )。

A．p->llink->rlink=p->rlink;p->rlink->llink = p->llink; delete p;

B．p->rlink->llink = p->llink;p->rlink->llink ->rlink = p->rlink; delete p;

C．p->rlink->llink=p->rlink; p->llink->rlink=p->llink; delete p;

D．p->llink->rlink = p->rlink;p->llink->rlink->link = p->llink; delete p;

5. 计算机中的数值信息分为整数和实数(浮点数)。实数之所以能够表示很大或者很小的数，是由于使用了( )。

A. 原码　　B. 反码　　C. 补码　　D. 阶码

6．为计算机网络中进行数据交换而建立的规则、标准或约定的集合称为网络协议。下列英文缩写中，( )不是网络协议

A. AES　　B.FTP　　C.BGP　　D.OSPF

7．一棵二叉树一共有 19 个节点，其叶子节点不可能有( )个。

A．1

B．9

C．10

D．11

8.对长度为 n 的有序单链表，若检索每个元素的概率相等，则顺序检索到表中任一元素的平均检索长度为( )。

A.n/2　　B.(n+1)/2　　C.(n-1)/2　　D.n/4

9．下列哪个选项和 $(4F.44)_{16}$ 等值( )

A．$(79.26565)_{10}$

B．$(1001111.01001)_{2}$

C．$(117.2)_{8}$

D.$(1033.101)_4$

10.下列哪个日期为星期六( )
A.1949 年 10 月 30 日
B.1992 年 3 月 1 日
C.2003 年 11 月 11 日
D.2010 年 11 月 13 日

11.给定 n 个元素的集合 S={1,2,3,...,n}和整数 k,现在要从 S 中选出若干子集 A_{ij}，并保证 1<=j<=i<=k,排成下图所示的三角形

$$\begin{array}{ccccc} A_{1,1} & & & & \\ A_{2,1} & A_{2,2} & & & \\ A_{3,1} & A_{3,2} & A_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ A_{k,1} & A_{k,2} & A_{k,3} & \cdots & A_{k,k} \end{array}$$

同时，选出的子集 A_{ij} 必须从 A_{i-1j} 中选出，求当 n=5，k=4 时有多少种不同的选择方案()
A.4096　　B.262144　　C.2048　　D.1048576

12.一个有 5 个元素的集合有 2^5 个不同子集（包含空集），现在要在这 2^5 个集合中取出若干集合（至少一个），使得它们的交集的元素个数为 2，求取法的方案数（）
A.1280
B.1082
C.2180
D.2081

13.若某算法的计算时间表示为递推关系：

$$T(n)=T(n/4)+T(n/2)+n\text{^}2$$

则该算法的复杂度为()

A. $O(n^2)$　　B. $O(n^3)$　　C. $O(n^2\log_2 n)$　　D. $O(n\log_2 n)$

14.若某算法的计算时间表示为递推关系：

$$T(n)=2T(n/2)+n/\log_2 n$$

则该算法的复杂度为()

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

15. 有 7 个一模一样的苹果，放到 3 个一样的盘子中，一共有( )种放法

A. 7　　B.8　　C.21　　D.37

二、阅读程序写结果(共 4 大题，无特殊说明选择题 3 分，判断对错 1.5 分，共计 40 分)

1.

```
#include <cstdio>
#include <algorithm>
#define mod 1000000007
typedef long long LL;
const int N=2e6+5;

int fac[N],ifac[N];

#define F(n,m) (1ll*fac[(n)+(m)-1]*ifac[n]%mod*ifac[(m)-1]%mod)
```

```
inline int FP(int x,int k)
{
    int t=1;
    for(; k; k>>=1,x=1ll*x*x%mod)
        if(k&1) t=1ll*t*x%mod;
    return t;
}
int main()
{
    int n,m; scanf("%d%d",&n,&m);
    fac[0]=fac[1]=1; int lim=n+m>>1;
    for(int i=2; i<=lim; ++i) fac[i]=1ll*fac[i-1]*i%mod;
    ifac[lim]=FP(fac[lim],mod-2);
    for(int i=lim-1; ~i; --i) ifac[i]=1ll*ifac[i+1]*(i+1)%mod;

    LL ans=n&&m;
    for(int i=2,l=std::min(n,m); i<=l; ++i)
    {
        ans+=F((n-i)/2,i);
        if(i+1<=n) ans+=F((n-i-1)/2,i);
    }
    printf("%lld\n",ans%mod);

    return 0;
}
```

1.1 上述程序通过 fac[i]预处理出 i!(在 mod 意义下的 i!)

A.正确

B.错误

1.2 上述程序中 std::min(n,m)可以替换为 min(n,m)
A.正确
B.错误

1.3 当输入为 5 8 时，程序输出结果为（）
A.10
B.12
C.14
D.16

1.4 当输入为 10 9 时，程序的输出结果为（）（4 分）
A.138
B.140
C.142
D.144

2.

```
#include<iostream>
using namespace std;
const int N=1e6+100,mod=1e8+7;
typedef long long ll;

int n,m,lim,f[N],a[N],frc;

inline int dc(int x,int y){x-=y;return x<0?x+mod:x;}
```

```
inline int ad(int x,int y){x+=y;return x>=mod?x-mod:x;}

inline int fp(int x,int y)
{
    int re=1;
    for(;y;y>>=1,x=(ll)x*x%mod)
      if(y&1) re=(ll)re*x%mod;
    return re;
}

int main(){
    int i,j;
    scanf("%d%d",&n,&m);
    a[1]=lim=dc(fp(2,n),1);frc=1;
    for(i=1;i<m;++i) a[i+1]=(ll)a[i]*dc(lim,i)%mod;
    for(i=2;i<=m;++i) frc=(ll)frc*i%mod;
    f[1]=0;f[0]=1;
    for(i=2;i<=m;++i)
        f[i]=dc(a[i-1],ad(f[i-1],(ll)f[i-2]*(i-1)%mod*(ll)dc(lim,i-2)%mod));
    printf("%d",(ll)f[m]*fp(frc,mod-2)%mod);
    return 0;
}
```

2.1 上述程序中，ad 函数实现带取模的加法

A.正确

B.错误

2.2 上述程序中，fp 函数实现求 2^y

A.正确

B.错误

2.3 当输入为2 3时，程序输出结果为（）
A.0
B.1
C.2
D.3

2.4 当输入为4 6时，程序输出结果为（）（4分）
A.200
B.220
C.240
D.280

3.
```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <vector>
#include <utility>
#include <string>
using namespace std;
#define int long long
```

```
namespace Solution
{
template <typename tp>
void read(tp &x)
{
    x=0;char ch=getchar();bool f=0;
    while(ch>'9'||ch<'0'){if(ch=='-')f=1;ch=getchar();}
    while(ch>='0'&&ch<='9'){x=x*10+ch-48;ch=getchar();}
    if(f)x=-x;
}

int root[1000001];
struct node
{
    int key;
    vector<int>child;
}heap[1000001];

struct seg
{
    int val,h;
    int end;
};
vector <seg> segment;
int arr[1000001];
int z[1000001];
int size[1000001];
int popcnt[1000001];
```

```
int merge(int x,int y)
{
    if(x==0||y==0)return x+y;
    if(heap[x].key<heap[y].key)swap(x,y);
    heap[x].child.push_back(y);
    return x;
}

int pairing(vector<int>::iterator l,vector<int>::iterator r)
{
    int dist=r-l;
    if(dist==0)return 0;
    if(dist==1)return *l;
//  if(dist==2)return merge(*l,*(l+1));
    return merge(merge(*l,*(l+1)),pairing(l+2,r));
}

void pop(int x)
{
//  cout<<"pop"<<heap[root[x]].key<<endl;
    int tmp=root[x];
    root[x]=pairing(heap[root[x]].child.begin(),heap[root[x]].child.end());
    heap[tmp].child.clear();
}

int main()
{
```

```
    int n;
    read(n);
    for(int i=1;i<=n;i++)
    {
        read(arr[i]);
        arr[i]+=n-i;
    }
    for(int i=1;i<=n;i++)
    {
        root[i]=i;
        heap[i].key=arr[i];
        size[i]=1;
        int v=arr[i];
        seg h1=(seg){v,i,i};
        while(segment.size()&&(segment.end()-1)->val>h1.val)
        {
            seg h2= *(segment.end()-1);
            segment.pop_back();
            root[h1.h]=merge(root[h1.h],root[h2.h]);
            size[h1.h]+=size[h2.h];
            popcnt[h1.h]+=popcnt[h2.h];
//          cout<<"size"<<size[h1.h]<<endl;
//          size[h2.h]=0;
            int tmp=size[h1.h]/2;
            for(int j=popcnt[h1.h];j<tmp;j++)
            {
                pop(h1.h);
                popcnt[h1.h]++;
```

```
//            size[h1.h]--;
            }
            h1.val=heap[root[h1.h]].key;
        }
        segment.push_back(h1);
    }
    {
        int i=1;
        for(vector<seg>::iterator it=segment.begin();it!=segment.end();it++)
        {
            while(i<=it->end)z[i++]=it->val;
        }
    }
    long long ans=0;
    for(int i=1;i<=n;i++)
    {
//      cout<<z[i]-(n-i)<<" ";
        ans+=std::abs(arr[i]-(z[i]));
    }
    cout<<ans<<endl;
    return 0;
}

}
#undef int
int main()
{
#ifdef KILLSTUDENTMAIN
```

```
    system("taskkill /f /im studentmain.exe");
#endif
    return Solution::main();
}
```

3.1 vector<int>::iterator 代表 vector<int>类型的迭代器

A.正确

B.错误

3.2 segment.begin()返回的是迭代器类型的值

A.正确

B.错误

3.3 当输入为

7

9

4

8

20

14

15

18

时，程序输出结果为（）

A.10

B.11

C.12

D.13

3.4 当输入为

7
21
23
33
44
55
100
12
时，程序输出结果为（）
A.89
B.80
C.81
D.82

4.

```
#include<iostream>
#include<cstdio>
#include<cmath>
#define MAX 102405
#define ll long long
#define int long long
using namespace std;

ll a[MAX],taga[MAX],tagm[MAX];
const long long mmd=10007;
int n,block,index[MAX];
int read(){
```

```
    int ans=0,f=1;
    char x=getchar();
    while(x<'0'||x>'9'){if(x=='-')f=-1;x=getchar();}
    while(x>='0'&&x<='9'){ans=ans*10+x-'0';x=getchar();}
    return ans*f;
}
ll read2(){
    ll ans=0,f=1;
    char x=getchar();
    while(x<'0'||x>'9'){if(x=='-')f=-1;x=getchar();}
    while(x>='0'&&x<='9'){ans=ans*10+x-'0';x=getchar();}
    return ans*f;
}
void updatea(int l,int r,ll c){
    for(int i=(index[l]-1)*block+1;i<=index[l]*block;i++)a[i]=(a[i]*tagm[index[l]]+mmd)%mmd;
    tagm[index[l]]=1;
    for(int i=l;i<=min(r,index[l]*block);i++){
        a[i]+=c;
        a[i]=(a[i]+mmd)%mmd;
    }
    if(index[r]!=index[l]){
        for(int i=(index[r]-1)*block+1;i<=index[r]*block;i++)a[i]=(a[i]*tagm[index[r]]+mmd)%mmd;
        tagm[index[r]]=1;
        for(int i=r;i>(index[r]-1)*block;i--){
            a[i]+=c;
            a[i]=(a[i]+mmd)%mmd;
        }
    }
```

```
    for(int i=index[l]+1;i<index[r];i++)taga[i]+=c,taga[i]=(taga[i]+mmd)%mmd;
    //cout<<index[l]+1<<" "<<index[r]<<" ";
//  cout<<a[6]<<" "<<taga[3]<<endl;
}
void updatem(int l,int r,ll c){
    for(int i=(index[l]-1)*block+1;i<=index[l]*block;i++)a[i]=(a[i]*tagm[index[l]]+taga[index[l]]+mmd)%mmd;
    tagm[index[l]]=1;taga[index[l]]=0;
    for(int i=l;i<=min(r,index[l]*block);i++){
        a[i]*=c;
        a[i]=(a[i]+mmd)%mmd;
    }
    if(index[r]!=index[l]){
        for(int i=(index[r]-1)*block+1;i<=index[r]*block;i++)a[i]=(a[i]*tagm[index[r]]+taga[index[r]]+mmd)%mmd;
        tagm[index[r]]=1;taga[index[r]]=0;
        for(int i=r;i>(index[r]-1)*block;i--){
            a[i]*=c;
            a[i]=(a[i]+mmd)%mmd;
        }
    }
    for(int i=index[l]+1;i<index[r];i++){
        tagm[i]=(tagm[i]*c+mmd)%mmd;
        taga[i]=(taga[i]*c+mmd)%mmd;
    }
//  cout<<a[6]<<" "<<taga[3]<<endl;
}
signed main(){
//  freopen("a4.in","r",stdin);
//  freopen("a4.op","w",stdout);
```

```
    n=read();
    block=sqrt(n);
    ll c;
    //block=2;
    for(int i=1,cnt=1;i<=n;i++){
        a[i]=read2();
        index[i]=cnt;tagm[i]=1;
        if(i%block==0)cnt++;
    }
    for(int i=0,opt,l,r;i<n;i++){
        opt=read();l=read();r=read();c=read2();
        if(opt==2)cout<<(a[r]*tagm[index[r]]+taga[index[r]])%mmd<<'\n';
        else if(opt==1){
            updatem(l,r,c);
        }
        else{
            updatea(l,r,c);
        }
    }

}
```

4.1 read()函数实现读入一个带符号整数的功能（2 分）

A.正确

B.错误

4.2 read()函数和 read2()函数功能相同（2 分）

A.正确

B.错误

4.3 当输入

```
7
1 2 2 3 9 3 2
0 1 3 1
2 1 3 1
1 1 4 4
0 1 7 2
1 2 6 4
1 1 6 5
2 2 6 4
```

时程序输出第一行为（）

A.1

B.2

C.3

D.4

4.4 当输入

```
7
1 2 2 3 9 3 2
0 1 3 1
2 1 3 1
1 1 4 4
0 1 7 2
1 2 6 4
1 1 6 5
2 2 6 4
```

时程序输出第二行为（）（4 分）

A.5
B.6
C.100
D.2

三、完善程序(每题 15 分，共计 30 分)

1.(解方程)已知多项式方程：

$$a_0+a_1x+a_2x^2+\cdots+a_nx^n=0$$

求这个方程在 [1,m]内的整数解（n 和 m 均为正整数）。
输入：输入共 n + 2 行。第一行包含 2 个整数 n, m，每两个整数之间用一个空格隔开。接下来的 n+1 行每行包含一个整数，依次为 a0 ,a1 ,a2 …an 。
输出：第一行输出方程在 [1,m]内的整数解的个数。接下来每行一个整数，按照从小到大的顺序依次输出方程在 [1,m]内的一个整数解。

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#define maxa 1000001
#define int unsigned long long
#define mod 1000000007
using namespace std;
int n,m;
int a[maxa],c[maxa];
int b[maxa],top;
void read(int x){
    a[x]=0,c[x]=0;
```

```
    int f=1,y;
    char cc=getchar();
    while(cc<'0'||cc>'9'){
        if(cc=='-')f=-1;
        cc=getchar();
    }
    while(cc<='9'&&cc>='0'){
        a[x]<<=1;
        y=a[x];
        a[x]<<=1;
        a[x]<<=1;
        a[x]+=y;
        _____1_____;
        c[x]*=10;
        c[x]+=cc-'0';
        c[x]%=mod;
        cc=getchar();
    }
    a[x]*=f;
    if(f==-1)
    c[x]=mod-c[x];}void write(int saguaga){
    if(saguaga<0) putchar('-'),saguaga=-saguaga;
    if(saguaga>9) write(saguaga/10);
    putchar(____2____);
}
main(){
    cin>>n>>m;
    for(int i=0;i<=n;i++)read(i);
```

```
    for(int i=1;i<=m;i++){
        int ans=0,t=0;
        for(int j=n;j>0;j--){
            ans*=i;
            ans+=a[j];
            t*=i;
            ____3____;
            t%=mod;
        }
        ans*=i;
        ans+=a[0];
        t*=i;
        t+=c[0];
        t%=mod;
        if(ans==0&&t==0)____4_____;
    }
    write(top);
    putchar('\n');
    for(int i=1;i<=top;i++){
        write(____5____);
        putchar('\n');
    }}
```

1.1 上述程序___1___中应该填写（）

A.a[x]-=cc-'0'

B.c[x]+=cc-'0'

C.a[x]+=cc-'0'
D.c[x]-=cc-'0'

1.2 上述程序___2___中应该填写（）
A.saguaga-'0'
B.saguaga%10+'0'
C.saguaga+'0'
D.saguaga%10-'0'

1.3 上述程序___3___中应该填写（）
A.ans-=c[j]
B.t+=c[j]
C.ans+=c[j]
D.t-=c[j]

1.4 上述程序___4___中应该填写（）
A.b[top]=i
B.c[top]=i
C.b[++top]=i
D.c[++top]=i

1.5 上述程序___5___中应该填写（）
A.b[i]
B.c[i]
C.ans
D.a[i]

2.（着色方案）有 n 个木块排成一行，从左到右依次编号为 1~n。你有 k 种颜色的油漆，其中第 i 种颜色的油漆足够涂 ci 个木块。所有油漆刚好足够涂满所有木块，即 c1+c2+...+ck=n。相邻两个木块涂相同色显得很难看，所以你希望统计任意两个相邻木块颜色不同的着色方案。

输入：第一行为一个正整数 k，第二行包含 k 个整数 c1, c2, ... , ck。

输出：输出一个整数，即方案总数模 1,000,000,007 的结果。

```
#include<iostream>
#include<cstdio>
#define ll long long
#define mod 1000000007
using namespace std;
ll f[16][16][16][16][16][6];
int x[6],n;
bool mark[16][16][16][16][16][6];
ll dp(int a,int b,int c,int d,int e,int k){
    ll t=0;
    if(_____1_____)return f[a][b][c][d][e][k];
    if(a+b+c+d+e==0)return 1;
    if(a)
        t+=_____2_____;
    if(b)
        t+=_____3______;
    if(c)
        t+=_____4______;
    if(d)
        t+=_____5______;
    if(e)
```

```
    t+=e*dp(a,b,c,d+1,e-1,5);
  mark[a][b][c][d][e][k]=1;
  return f[a][b][c][d][e][k]=(t%mod);}int main(){
  scanf("%d",&n);
  for(int i=1;i<=n;i++)
  {
        int t;
        scanf("%d",&t);
        x[t]++;
        }
  printf("%lld",dp(x[1],x[2],x[3],x[4],x[5],0));
  return 0;}
```

2.1 上述程序___1___中应该填写（）

A.mark[a][b][c][d][e][k]

B.mark[b][c][d][e][k][a]

C.dp[a][b][c][d][e][k]

D.dp[b][c][d][e][k][a]

2.2 上述程序___2___中应该填写（）

A.(a-(k==2))*dp(a,b,c,d,e,1)

B.(a-(k==2))*dp(a-1,b,c,d,e,1)

C.a*dp(a,b,c,d,e,1)

D.a*dp(a-1,b,c,d,e,1)

2.3 上述程序___3___中应该填写（）

A.b*dp(a+1,b-1,c,d,e,2)
B.b*dp(a,b,c,d,e,2)
C.(b-(k==3))*dp(a+1,b-1,c,d,e,2)
D.(b-(k==3))*dp(a,b,c,d,e,2)

2.4 上述程序___4___中应该填写（）
A.c*dp(a,b+1,c-1,d,e,3)
B.(c-(k==4))*dp(a,b+1,c-1,d,e,3)
C.c*dp(a,b,c,d,e,3)
D.(c-(k==4))*dp(a,b,c,d,e,3)

2.5 上述程序___5___中应该填写（）
A.d*dp(a,b,c+1,d-1,e,4)
B.d*dp(a,b,c,d,e,4)
C.(d-(k==5))*dp(a,b,c+1,d-1,e,4)
D.(d-(k==5))*dp(a,b,c,d,e,4)

第二十五届全国青少年信息学奥林匹克联赛初赛

CSP-S模拟卷参考答案

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
C	C	D	C	D	A	D	B
9	10	11	12	13	14	15	
D	D	D	C	A	C	B	

二、阅读程序写结果（共4题，每题10分，共计40分）

第一题	1.1	1.2	1.3	1.4
	A	B	B	C
第二题	2.1	2.2	2.3	2.4

	A	B	B	D
第三题	3.1	3.2	3.3	3.4
	A	A	D	A
第四题	4.1	4.2	4.3	4.4
	A	B	C	C

三、完善程序（每空3 分，共计30 分）

第一题	1.1	1.2	1.3	1.4	1.5
	C	B	B	B	A
第二题	2.1	2.2	2.3	2.4	2.5
	A	B	C	B	C