1. 选择题（每题2分，共计30分）

1、与十进制小数17.5625等值的八进制小数为（B）。

A、21.5625 B、21.44 C、21.73 D、21.731

解析： 5\*1/8+6\*1/64+...

2、设待排序的记录为(49,38,65,97,76,13,27,49,55,4)，经过下面过程将序列排序：

第一趟：13,27,49,55,4,49,38,65,97,76

第二趟：13,4,49,38,27,49,55,65,97,76

第三趟：4,13,27,38,49,49,55,65,76,97

问它所用的方法是：（D）。

A、冒泡排序 B、直接选择排序

C、基数排序 D、希尔排序

解析：49 13 38 27… 先是距离为5的五组，排完序变为第一趟，然后是3

3、下面是一段C++程序：

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

int x=A[h+1];

int k=h;

while(k>=1 && A[k]>x){

A[k+1]=A[k];k--;

}

A[k+1]=x;

}

假设在程序开始执行时，数组A[1…n]是按照关键字非递减有序排列时，下列答案中，（B）最好地描述了最好情况下的程序排序的时间复杂度。

A、O(nlogn) B、O(n)

C、O(logn) D、O(n2)

解析：插排，最好情况为序列有序

4、给定一个正整数N=8934632178，现决定依次删除其中6个数位上的数字（每次删除一个数位上的数字），每次删除后按原来的次序组成一个新数M的值均是当前状态下的最小数，则第四次应该删除的数字是（D）。

A、6 B、8 C、7 D、4

解析：贪心。从高位开始搜索递减区间。 9 8 6 4...

5、下列形状的三角形中，字幕a-i分别表示数字1,2,3,…,9

a

b c

d e

f g h i

字母a-i同时满足下列条件：

1. a<f<i
2. b<d,g<h,c<e
3. a+b+d+f=f+g+h+i=i+e+c+a=19

则满足条件的三角形个数为（C）。

A、2 B、3 C、4 D、5

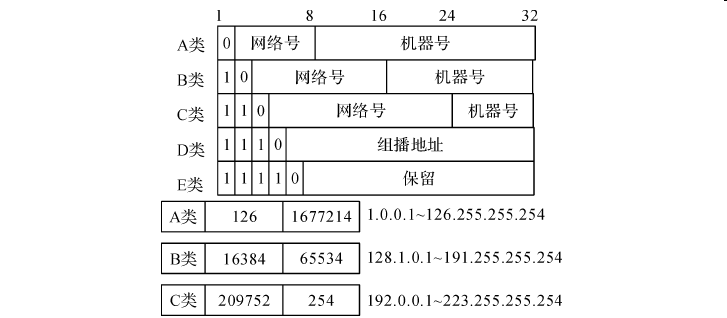
6、在微机系统中，最基本的输入输出模块BIOS存放在（B）中。

A、RAM B、ROM C、硬盘 D、寄存器

7、下列地址中，属于B类IP地址的是（C）。

A、27.33.119.2 B、192.97.32.121 C、133.201.189.32 D、126.33.82.107

解析：



8、一棵二叉树的高度为h（根为第一层），所有节点的度为0或2，则此树最少有（B）个节点。

1. 2h-1 B、2h-1 C、2h+1 D、h+1

9、下列关于排序说法不正确的是（D）。

1. 插入排序、冒泡排序是稳定的
2. 选择排序的时间复杂度为O(n2)
3. 选择排序、希尔排序、快速排序、堆排序是不稳定的
4. 希尔排序、快速排序、堆排序的时间复杂度为O(nlogn)

解析：是否稳定看排序后原来相等两数是否会交换位置，因此，如果不是相邻两数比较交换的往往是不稳定。shell是O(n^1.3),一般达不到O(nlogn)

10、假设我们用d=(a1,a2,…,a5)，表示无向图G的5个顶点的度数，下面给出的哪（些）组d值合理（B）。

A、{5,4,4,3,1} B、{4,2,2,1,1} C、{3,3,3,2,2} D、{2,2,2,2,3}

解析：无向图总度数为偶数

11、若已知一个栈的入栈顺序1,2,3,…,n，其输出序列为P1,P2,P3,…,Pn（它是输入序列的一个排序），则在输出序列中可能出现的情况是（A）。

A、Pj<Pk<Pi,其中i<j<k B、Pk<Pj<Pi,其中i<j<k

C、Pj<Pi<Pk,其中i<j<k D、Pi<Pk<Pj,其中i<j<k

12、设有一个含有13个元素的Hash表（0-12），Hash函数是：H(key)=key%13，其中%是求余数运算。用二次探查法解决冲突，则对于序列(8,31,20,33,18,53,27)，则下列说法不正确的是（D）。

A、18在4号格子中 B、33在6号格子中

C、31在5号格子中 D、20在6号格子中

解析：二次探查法：+1 -1 +4 -4 +9 -9…

13、任意一棵树均可唯一地转换成与它对应的二叉树。由树转换成的二叉树中，节点N的左右子节点分别是N在原树里对应节点的（A）。

A、最左子节点/最邻近的右兄弟 B、最右子节点/最右的兄弟

C、最邻近的右兄弟/最左的兄弟 D、最邻近的左兄弟/最邻近的右兄弟

14、关于有权值的无向图最小生成树，以下说法正确的是（B）。

A、一个图的最小生成树是唯一的

B、一个图的最小生成树不一定唯一，但任意两个的最小生成树的第K大的边大小相同。

C、一个图的最小生成树中不一定包含该图中权值最小的边

D、一个图的最小生成树中一定不包含该图中权值最大的边。

解析：最小生成树有最优子结构，所以B对；无法避免情况下，也可能选到权值最大边，D不对。

15、已知6个结点的二叉树的先序遍历是1 2 3 4 5 6（数字为结点编号），后序遍历是3 2 5 6 4 1，则该二叉树的可能的中序遍历是（B）。

A、3 2 1 4 6 5

B、3 2 1 5 4 6

C、2 3 5 1 4 6

D、2 3 1 4 6 5

解析：已知先（后）序中序可唯一确定。

二、阅读程序写结果（共4题，每题8分，共计32分）

1、

#include<iostream>

using namespace std;

int a,b,c,d,e;

int main() {

a=1;b=a+2;c=b+3;d=c+4;a=d+5;

for(int e=1;e<=4;e++){

if(a%2==0) a=a/2+d;

else a=a/2+d+1;

b=a+2;c=b+3;d=c+4;a=d+5;

}

cout<<a<<endl;

return 0;

}

16、for循环中int可以去掉而不影响结果。（ ） 是

17、else改为if(a%2!=0)结果不变。（ ） 否

18、对于任意a的初始值，程序运行中a单调递增。 （ ）是

19、若初始a增加1，答案也增加1。 （ ）否

20、从第5行a=1之后，程序运行过程中，a总共变化了几次？（ ） D

A．7 B．8 C．9 D．10

21、程序输出结果为（ ）。 B

A．146 B．152 C．160 D．252

2、

#include<iostream>

#include<iomanip>

using namespace std;

int a[33],b[33];

void ssort(int x,int y){

if(y-x>1){

int m=(x+y)/2;

ssort(x,m);

ssort(m+1,y);

int k=x;

for(int i=x;i<=m;i++){

b[k]=a[i];b[k+1]=a[m+i-x+1];

k+=2;

}

for(int i=x;i<=y;i++) a[i]=b[i];

}

}

int main() {

for(int i=1;i<=16;i++) a[i]=i;

ssort(1,16);

for(int i=1;i<=16;i++) cout<<setw(3)<<a[i];

cout<<endl;

return 0;

}

22、去掉第二行不影响程序运行。（ ） 否

23、k的值域为[1,13]。（ ） 否

24、结果中a[i]数值均不同。 （ ）是

25、若第7行m=(x+y)/2改为m=(x+y+1)/2,答案会发生变化。 （ ）是

26、若16改为n，该程序的时间复杂度为？（ ） B

A．O(n) B．O(nlogn) C．O(nlog^2 n) D．O(n^2)

主定理可解：T(n)=T(n/2)+O(n)

27、程序输出结果为（ ）。 B

这道题请严格按照输出格式来，注意行首和行间空格，行末空格忽略

A．1 9 5 13 3 11 7 15 2 10 6 14 4 12 8 16

B． 1 9 5 13 3 11 7 15 2 10 6 14 4 12 8 16

C．5 9 1 13 3 11 7 15 6 10 2 14 4 12 8 16

D． 5 9 1 13 3 11 7 15 6 10 2 14 4 12 8 16

3、

#include<iostream>

using namespace std;

int w[10];

int n,m,weight;

int main() {

cin>>weight;cout<<weight<<'=';

m=1;int i=0;w[i]=1;

while(m<weight){

i++;w[i]=w[i-1]\*3;m+=w[i];

}

bool flg=false;

n=weight;i=0;

while(n>0){

switch(n%3){

case 0:n/=3;break;

case 1:

n/=3;if(flg) cout<<'+';

cout<<w[i];flg=true;break;

case 2:

n=n/3+1;cout<<'-'<<w[i];

flg=true;break;

}

i++;

}

return 0;

}

解析：将一个数用几个3次幂通过+-连接。25=2\*3^2+2\*3+1\*3^0 221

我们可以分析，任意一个三次幂有三种状态：+、-、不取，那么我们可以表示为-1,0,1三种状态，这样自然想到用三进制表示，但是三进制数码为0,1,2，与实际相比每位都大1，因此可以将原数三进制拆分后各位-1,如25=2\*3^2+2\*3+1\*3^0，三进制为221，各位-1得110表示+、+、不取。

28、第8行while循环结束时m是最接近weight的三次幂之和。（ ） 否

29、第8行while循环中’<’改成’<=’不会影响程序结果。（ ） 是

30、case2中的break去掉不会影响程序结果。 （ ）是

31、[1,1000]范围内的weight，均能完成拆分。（ ） 是

32、weight为928，结果会包含多少个不同的三进制数？（ ） B

A．5 B．6 C．7 D．8

33、若输入518，程序输出结果为（ ）。 A

A．518=-1-3+9+27-243+729

B．-1-3+9+27-243+729

C．518=729-243+27+9-3-1

D．729-243+27+9-3-1

完善程序（共2题11空，每题后三空为3分，其它一空2分，共28分）

1. 现在政府计划在某个区域的城市之间建立告诉公路，以使得其中任意两个城市之间都有直接或间接的告诉公路相连。费用为每千米一个单位价格，求最小费用。

输入：n（n<=100，表示城市数目）

接下来n行每行两个数xi，yi，表示第i个城市的坐标。（单位：千米）

输出：最小费用（保留两位小数）

解析：prim。4,6空较难，容易填写不全，注意细节，不要忘了加上p[j]>0。

#include<iostream>

#include<iomanip>

#include<cmath>

using namespace std;

const int MaxN=101;

struct Tcity{

double x,y;

}c[MaxN];

double d[MaxN][MaxN];

int p[MaxN];

inline double cal(double a){return (1);}

int main() {

int n;

cin>>n;

for(int i=1;i<=n;i++) cin>>c[i].x>>c[i].y;

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

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

d[i][j]=sqrt(cal(c[i].x-c[j].x)+cal( (2) ));

p[1]=0;

for(int i=2;i<=n;i++) (3);

double ans=0;

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

double Min=1e10;

int k;

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

if( (4) ){Min=d[p[j]][j]; (5) ;}

ans+=d[p[k]][k];

p[k]=0;

for(int j=1;j<=n;j++) if( (6) )p[j]=k;

}

cout<<fixed<<setprecision(2)<<ans<<endl;

return 0;

}

34、1、2处填（ ）。D

A．0 c[i].x-c[j].y B．1 c[i].x-c[j].x

C. a c[i].x-c[j].y D．a\*a c[i].y-c[j].y

35、3处填（ ）。A

A . p[i]=1 B．p[i]=i

C. p[i]=MaxN D．p[i]=i\*i

36、4处填（ ）。C

1. p[j]>0 B．d[p[j]][j]<Min
2. p[j]>0&&d[p[j]][j]<Min D．p[j]>0||d[p[j]][j]<Min

37、5处填（ ）。B

A . k++ B．k=j

C.k=Min D．Min++

38、6处填（ ）。D

A . d[k][j]<d[p[j]][j] B．p[j]>0

C．p[j]>0||d[k][j]<d[p[j]][j] D．p[j]>0&&d[k][j]<d[p[j]][j]

2、给定一个01字符串，请你找出长度介于a，b之间，重复出现次数最多的01串。

输入：a，b(0<=A<=B<=12)

由0、1组合的序列，由“.”结尾。

输出：重复出现的最多次数以及所有满足要求的串。

提示：本程序中将01序列转换为2进制存储。

#include<iostream>

using namespace std;

int m[8193],two[21],v[21];

int main() {

for(int i=1;i<=13;i++) two[i]= (1) ;

int a,b;

cin>>a>>b>>c;

int s=1,k=1;

while(c!='.'){

s=(s<<1)+c-'0';

if( (2) )s=(s-two[b+1])%two[b]+two[b];

m[s]++;

if(k<b)

for(int i=a;i<k;i++) (3) ;

k++;

cin>>c;

}

for(int i=two[b];i<=two[b+1];i++)

if(m[i]>0)

for(int j=a;j<b;j++) m[i%two[j]+two[j]]= (4) ;

int Max=0;

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

if(m[i]>Max) Max=m[i];

cout<<Max<<endl;

for(int i=two[a];i<=(5);i++)

if(m[i]==Max){

int j=0,k=i;

do{

j++;v[j]=k%2;k/=2;

}while( k!=1 );

while(j>0){cout<<v[j];j--;}

cout<<endl;

}

return 0;

}

解析：每个01串都用二进制表示，但是0011和011这两个不同的串在二进制下都是11，为了区分，在每个二进制前面加上一个1，则上述两个分别表示为10011,1011.程序思路是：所有长度小于b的串都包含于长度等于b的串中。因此程序在读入时先将所有长度等于b的串记录下来，然后再去计算长度小于b的串。

39、1处填（ ）。B

A．i B．1<<i

C. i+1 D．1<<(i+1)

40、2处填（ ）。C

A . s>two[b+1] B．k<=b

C. s>=two[b+1] D．k<b

41、3处填（ ）。D

1. m[s%two[i]]]+two[i]++ B．m[s%two[i]]+two[i]
2. m[s%two[i]+two[i]++] D．m[s%two[i]+two[i]]++

42、4处填（ ）。D

A . m[i] B． two[j]]+m[i]

C. m[i%two[j]]+ m[i] D．m[i%two[j]+two[j]]+m[i]

43、5处填（ ）。A

A . two[b+1] B．two[a+1]

C．Max D．13