1. [单选题]

当分配给一个进程的页面数增加时，页故障数可能增大也可能变小，下述算法符合这种情况的是

FIFO算法

LRU算法

Clock算法

LFU算法

2. [单选题]

一个提供NAT服务的路由器在转发一个源IP地址为10.0.0.1、目的IP地址为131.12.1.1的IP分组时，可能重写的IP分组首部字段是

Ⅰ.TTL

Ⅱ.片偏移量

Ⅲ.源IP地址

Ⅳ.目的IP地址

仅Ⅰ

仅I、Ⅱ

仅Ⅰ、Ⅱ、III

Ⅰ、Ⅱ、Ⅲ、Ⅳ

3. [单选题]

在深度学习网络中， 以下哪种技术不是主要用来做网络正则化的（提升模型泛化能力）

dropout

参数共享

Early stopping

Pooling

4. [单选题]

[机器学习]以下不属于有监督的词义消歧方法的是

Flip-Flop算法

贝叶斯分类器

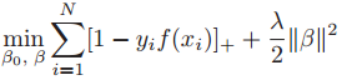
最大熵消歧

基于词典的消歧

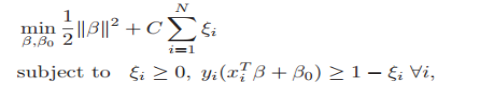
5.[单选题]

SVM的以下两种模型表达是等价的， 则其中的正则化系数λ和C的关系为：

形式一：



形式二：



λ=C

λ=1/C

λ=C的平方

6. [单选题]

若前缀表达式为-+a\*b-cd/ef，后缀表达式为abcd-\*+ef/-，那么对应二叉树的中序遍历序列是（）

a+c\*d-b-e/f

a+b\*c-d-e/f

a+b\*d-c-e/f

a+e\*c-d-b/f

7. [单选题]

对于字符串"ABCDADA"的二进制哈夫曼编码有多少位?

11

12

13

14

8. [单选题]

下面关于选择排序说法正确的是()

每扫描一遍数组，需要多次交换

选择排序是稳定的排序方法,因为时间复杂度是固定的O（n^2）

选择排序排序速度一般要比冒泡排序快

空间复杂度为O（1）

9. [单选题]

在UML建模中，下列哪个UML的图一般用于描述软件系统的需求()

状态图

协作图

用例图

顺序图

10.[编程题] 括号匹配深度

时间限制：1秒

空间限制：32768K

一个合法的括号匹配序列有以下定义:  
1、空串""是一个合法的括号匹配序列  
2、如果"X"和"Y"都是合法的括号匹配序列,"XY"也是一个合法的括号匹配序列  
3、如果"X"是一个合法的括号匹配序列,那么"(X)"也是一个合法的括号匹配序列  
4、每个合法的括号序列都可以由以上规则生成。  
例如: "","()","()()","((()))"都是合法的括号序列  
对于一个合法的括号序列我们又有以下定义它的深度:  
1、空串""的深度是0  
2、如果字符串"X"的深度是x,字符串"Y"的深度是y,那么字符串"XY"的深度为max(x,y) 3、如果"X"的深度是x,那么字符串"(X)"的深度是x+1  
例如: "()()()"的深度是1,"((()))"的深度是3。牛牛现在给你一个合法的括号序列,需要你计算出其深度。

**输入描述:**

输入包括一个合法的括号序列s,s长度length(2 ≤ length ≤ 50),序列中只包含'('和')'。

**输出描述:**

输出一个正整数,即这个序列的深度。

**输入例子1:**

(())

**输出例子1:**

2

11. [编程题] 奶牛编号

时间限制：1秒

空间限制：32768K

牛牛养了n只奶牛,牛牛想给每只奶牛编号,这样就可以轻而易举地分辨它们了。 每个奶牛对于数字都有自己的喜好,第i只奶牛想要一个1和x[i]之间的整数(其中包含1和x[i])。  
牛牛需要满足所有奶牛的喜好,请帮助牛牛计算牛牛有多少种给奶牛编号的方法,输出符合要求的编号方法总数。

**输入描述:**

输入包括两行,第一行一个整数n(1 ≤ n ≤ 50),表示奶牛的数量 第二行为n个整数x[i](1 ≤ x[i] ≤ 1000)

**输出描述:**

输出一个整数,表示牛牛在满足所有奶牛的喜好上编号的方法数。因为答案可能很大,输出方法数对1,000,000,007的模。

**输入例子1:**

4

4 4 4 4

**输出例子1:**

24

13.[单选题]

设置tcp的哪个socket参数会影响了 nagle算法？

TCP\_MAXSEG

TCP\_KEEPALIVE

TCP\_SYNCNT

TCP\_NODELAY

13. [编程题] 平方串

时间限制：1秒

空间限制：32768K

如果一个字符串S是由两个字符串T连接而成,即S = T + T, 我们就称S叫做平方串,例如"","aabaab","xxxx"都是平方串.

牛牛现在有一个字符串s,请你帮助牛牛从s中移除尽量少的字符,让剩下的字符串是一个平方串。换句话说,就是找出s的最长子序列并且这个子序列构成一个平方串。

输入描述:

输入一个字符串s,字符串长度length(1 ≤ length ≤ 50),字符串只包括小写字符。

输出描述:

输出一个正整数,即满足要求的平方串的长度。

输入例子1:

frankfurt

输出例子1:

4

答案：ACDDB

BCDC

10.#include<stdio.h>

#include<stack>

using namespace std;

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

int main(){

char s[100];

stack<char> stk;

int i,Max=0;

for(scanf("%s",s),i=0;s[i]!='\0';i++)

if(s[i]=='(') stk.push(s[i]),Max=max(Max,stk.size());

else stk.pop();

printf("%d",Max);

}

11.#include<stdio.h>

#include<algorithm>

using namespace std;

int main(){

int n,i,x,a[100];

long long res=1,mod=1000000007;

for(scanf("%d",&n),i=0;i<n;i++) scanf("%d",a+i);

for(sort(a,a+n),i=0;i<n;i++) res=res%mod\*(a[i]-i)%mod;

printf("%lld",res);

}

12.D

13.#include<iostream>

#include<string>

usingnamespacestd;

intMaxLength(string s,ints2)//从s2位置把字符串切开

{

intc[50][50]={0};

if(s2 >= s.size()) return0;

intsize1 = s2, size2 = s.size() - s2; for(inti = 1; i <= size1; i++)

{ for(intj = 1; j <= size2; j++){

if(s[i - 1] == s[s2 + j - 1]){

c[i][j] = c[i - 1][j - 1] + 1;

}

else{

c[i][j] = max(c[i - 1][j], c[i][j - 1]);

}

}

}

returnc[size1][size2];

}

int main()

{

string str;

cin>>str;

intmax=0;

intsize=str.size();

for(inti=0;i<size;i++)

{

inttmp;

tmp=MaxLength(str,i);

if(tmp>max)

max=tmp;

}

cout<<2\*max;

return0;

}