

快快编程模拟练习题

一、单项选择题（共 15 题，每题 2 分，共计 30 分，前 11 题为单选题，后 4 题为不定项选择题，全部选对才得分）

1. C++语言是一种（ ）

- A.高级语言 B.汇编语言 C.低级语言 D.机器语言

2. 与十进制数 17.5625 对应的 8 进制数是（ ）

- A. 21.5625 B.21.44 C.21.73 D.21.731

3. 字长为 32 位的计算机是指（ ）

- A.该计算机能够处理的最大数不超过 32
B.该计算机中的 CPU 可以同时处理 32 位的二进制信息
C.该计算机的内存量为 32MB
D.该计算机每秒所能执行的指令条数为 32MIPS

4. 在计算机中用黑白两色点阵来表示汉字字形的编码，一个 16*16 点阵字模的存储容量为（ ）

- A.128 字节 B. 16 字节 C. 32 字节 D. 64 字节

5. “网络通讯协议”，如：Internet 采用的 TCP/IP 等协议是一组（ ）

- A.软件 B.存储器 C.外部设备 D.约定的规则

6. 当 $A \geq B \& \& B \geq C$ 的取值为真时，那么表达式 $A > C || B == C$ 的值（ ）

- A.真 B.假
C.无法判断结果的真假 D. 只有当 A、B、C 都为正数时才为真

7. 从 1 到 n 这 n 各数中任取若干个数，不能取相邻的数，并且至少取一个，n=12 时，总共有（ ）种不同的取法

- A. 373 B. 374 C. 375 D. 376

8. 在所有三位数中，各位数字从高位到低位顺次减少的数共有（ ）个

- A. 110 B. 120 C. 130 D. 140

9. 6 名男生和 6 名女生围绕圆桌就坐,为了活跃气氛,男生和女生间隔而坐,每个男生左右两边都是女生,每个女生两边都是男生,共有（ ）种不同的就坐方案

- A. 120 B. 720 C. 86400 D. 665280

10. 已知 n 个顶点的有向图，若该图是强连通的（从所有顶点都存在路径到达其他顶点），则该图中最少有（ ）条有向边？

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

11. 对有序数组{5, 13, 19, 21, 37, 56, 64, 75, 88, 92, 100}进行二分查找, 等概率情况下, 查找成功平均查找长度(平均比较次数)是 ()

- A. 35/11 B. 34/11 C. 33/11 D. 32/11

12. (不定项) 关于有权值的无向图最小生成树, 以下说法正确的有 ()

- 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必然包括该图最小的边
D. 一个图的最小生成树中一定不包括该图中权值最大的边

13. (不定项) 下列有关算法的正确说法是 ()

- A. 算法必须有输出 B. 算法必须在计算机上用某种语言实现
C. 算法不一定有输入 D. 算法必须在有限步执行后能结束

14. (不定项) 对于一个大小为 3 的栈, 若输入顺序为 123456, 则以下出栈顺序有可能的是 ()

- A. 123456 B. 654321 C. 432165 D. 321654

15. (不定项) 下列 IP 地址中错误的是()

- A. 202.300.12.4 B. 192.168.0.3
C. 100:128:35:91 D. 111-102-35-21

二、阅读程序(程序输入不超过数组或字符串定义的范围: 判断题正确填√, 错误填×; 除特殊说明外, 判断题 1.5 分, 选择题 3 分, 共计 40 分)

1.

1	#include<iostream>
2	using namespace std;
3	int CycleNum(int i){
4	int count=1;
5	while(i!=1){
6	if(i%2)
7	i=3*i+1;
8	else
9	i/=2;
10	count++;
11	}
12	return count;
13	}
14	int main(){
15	int a,b;
16	cin>>a>>b;

17	int x=a<b?a:b;
18	int y=a>b?a:b;
19	int MaxCycle=0;
20	for(int i=x;i<=y;i++){
21	int temp=CycleNum(i);
22	if(MaxCycle<temp)
23	MaxCycle=temp;
24	}
25	cout<<MaxCycle<<endl;
26	return 0;
27	}

判断题:

1. 输入的第一个数据 a 必须小于第二个数据 b, 否则程序输出一定为 0。()
2. 程序输出的数值一定是一个输入的两个正整数 a, b 之间(包括 a 和 b)的正整数。()
3. 若将第 22 行的"MaxCycle<temp"改为"MaxCycle<=temp", 程序输出结果不变。()
4. 若输入的两组数据 a1,b1、a2,b2 有如下关系: a1<a2&&b1<b2 则前者的输出结果一定小于后者。()

选择题:

5. 若输入的数据为 1 5, 那么输出为 ()。
A.4 B. 5 C. 8 D.9
- 6.以下输出结果和 10 11 输出结果一致的是 ()。
A. 10 12 B. 9 12 C. 9 11 D. 8 12

2.

1	#include<iostream>
2	#include<algorithm>
3	using namespace std;
4	const int MMAX=20;
5	const int NMAX=50000;
6	const int INF=(1<<29);
7	int main(){
8	int n,m;
9	int C[21];
10	int T[NMAX+1];
11	cin>>n>>m;
12	for(int i=1;i<=m;i++){
13	cin>>C[i];
14	}
15	for(int i=0;i<=NMAX;i++) T[i]=INF;

16	T[0]=0;
17	for(int i=1;i<=m;i++){
18	for(int j=0;j<=n;j++){
19	if(j+C[i]<=n)
20	T[j+C[i]]=min(T[j+C[i]],T[j]+1);
21	}
22	}
23	cout<<T[n]<<endl;
24	}

完成下面的判断题和单选题

判断题:

1. 若输入的均为正整数，则输出一定大于等于 0。()
2. 若将第 16 行“T[0]=0;”改为“T[1]=0;”，程序输出结果一定不变。()
3. 若输入是 50 5 1 2 7 8 12，将第 17 行和 18 行互换位置，程序输出结果不变。()
4. 若输入的数据均为正整数，且 C 数组数值中包含 1，则若输出的数据一定小于等于输入的
的第一个数值 n。()

选择题:

5. 若输入的数据为 15 6 1 2 7 8 12 20，则输出为()
A.1 B. 2 C. 3 D. 4
6. 若输出的数据为 4，那么输入数据不可能是 ()。
A. 13 3 1 3 8 B. 4 1 1
C. 20 2 5 3 D. 16 4 2 4 10 11

3.

1	#include<iostream>
2	#include<cstring>
3	using namespace std;
4	const int maxn=101;
5	int ans[maxn][maxn],n;
6	void doit(){
7	memset(ans,0,sizeof(ans));
8	for(int i=1;i<=n;i++)ans[1][i]=1;
9	for(int i=2;i<=n;i++){
10	for(int j=1;j<=i;j++)
11	ans[i][j]=ans[i-j][j]+ans[i][j-1];
12	for(int j=i;j<=n;j++)
13	ans[i][j]=ans[i][j-1]+1;

14	}
15	}
16	int main(){
17	cin>>n;
18	doit();
19	cout<<ans[n][n]<<endl;
20	return 0;
21	}

判断题:

1. 将第 5 行和 6 行互换位置, 程序输出结果不变。()
2. 若程序两次输入不同的正整数数据, 则两次输出的结果一定不相同()

选择题:

3. 若输入为 5, 则输出为()。
A. 5 B. 6 C. 7 D. 8
4. 若输入为 4 则 a[4][5]的最终值为(); 若输入为 7 则 a[4][5]的最终值为()
A. 0、5 B. 0、0 C. 5、5 D. 15、15
5. 下列说法正确的是 ()
A. 若将第 8 行删去, 则无论输入何种数据, 程序必定输出 0;
B. 若将第 8 行“ans[1][i]=1”改写成“ans[1][i]=2”, 则无论输入何种数据, 程序输出相比原来一定变大
C. 若将第 9 行“i<=n”改写成“i<=2*n”, 则无论输入何种数据, 程序输出相比原来一定变小
D. 以上都不对
6. (4 分) 若输入为 10, 则输出为()。
A. 40 B. 41 C. 42 D. 43

三、完善程序 (单选题, 前 2 空 3 分, 3-6 空每题 2 分, 后四空每题 4 分, 共计 30 分)

1. (取石子) Alice 和 Bob 一起玩取石子的游戏, 石子有两堆, 每次取的时候至少取一颗, 可从任意一堆中取任意个, 或从两堆中取走相同多个, Alice 先手, 取完最后一颗石子的人获胜, 给出两堆石子的个数, 问 Alice 有无必胜策略?

样例输入: 1 2

样例输出: Bob wins

提示: 若一个状态有一个后继状态为必败态, 则该状态为必胜态, 相反, 若一个状态的所有后继状态都为必胜态, 则该状态为必败态。

1	#include <iostream>
2	using namespace std;

3	int a,b;
4	bool isok(int a,int b){//判断当前状态下先手是否胜利
5	int i,minab;
6	if(____(1)____) return 0;
7	for(i=1;i<=a;i++) if(!isok(a-i,b)) return 1;
8	for(i=1;i<=b;i++) if(____(2)____) return 1;
9	if(a<b)minab=a;else ____ (3) ____;
10	for(i=1;i<=minab;i++) if(! isok(a-i,b-i))return 1;
11	return ____ (4) ____;
12	}
13	int main(){
14	cin>>a>>b;
15	if(____(5)____) cout<<"Alice wins"<<endl;
16	else cout<<"Bob wins"<<endl;
17	return 0;
18	}

- (1) 处应填 ()。
A. a==0 B. b==0 C. a==0&&b==0 D. a==0||b==0
- (2) 处应填 ()。
A. !isok(a-i,b) B. !isok(a,b-i) C. !isok(a-i,b-i) D. !isok(a,b)
- (3) 处应填 ()。
A. minab=a-b B. minab=a C. minab=b D. minab=a+b
- (4) 处应填 ()。
A. minab B. isok(a,minab) C. 1 D.0
- (5) 处应填 ()。
A. isok(a,b) B. isok(b,a) C. isok(a-b,b) D. isok(b-a,a)

2. (自动反转) N 头牛排成了一列。每头牛或者向前或者向后。为了让所有牛都面向前方，农场管理员买了一台自动转向的机器。这个机器在购买时就必须设定一个数值 K，机器每操作一次恰好使 K 头连续的牛转向。请求出为了让所有牛都能面向前方需要的最少操作次数 M 和对应最小的 K。

【样例输入】

7

BBFBFB (F：面向前方，B：面向后方)

【样例输出】

3(K 值)

3(M 值)

(先反转 1-3 号的三头牛，然后再反转 3-5 号，最后反转 5-7 号)

试补全程序。

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int n;
5  int dir[5005]; //牛的方向 (0: F, 1: B)
6  int f[5005]; //区间[i, i+k-1]是否进行反转
7  //固定k, 求对应的最小操作回数
8  int cal(int k){
9      memset(f,0,sizeof(f));
10     int res=0,sum=0;
11     for(int i=0;i+k<=n;i++){ //计算区间[i,i+k-1]
12         if((dir[i]+sum)%2!=0){
13             res++;
14             f[i]=1;
15         }
16         sum+=____(1)____;
17         if(i-k+1>=0) sum-=f[i-k+1];
18     }
19     for(int i=____(2)____;i<n;i++){ //检查剩下的牛是否有朝后方的情况
20         if(____(3)____) //无解
21             return -1;
22         if(i-k+1>=0) sum-=f[i-k+1];
23     }
24     return ____ (4) ____;
25 }
26 void solve(){
27     int K=1,M=n;
28     for(int k=1;k<=n;k++){
29         int m=cal(k);
30         if(____(5)____){M=m;K=k;}
31     }
32     cout<<K<<" "<<M<<endl;
33 }
34 int main(){
35     cin>>n;
36     char ch;
37     for(int i=0;i<n;i++){
38         cin>>ch;
39         if(ch=='F')
40             dir[i]=0;
41         else
42             dir[i]=1;
43     }
```

44	<code>solve();</code>
45	<code>return 0;</code>
46	<code>}</code>

1. (1) 处应填 ()。

- A. i B. res C. f[i] D. f[i- 1]

2. (2) 处应填 ()。

- A. n-k+1 B. n-k C. 1 D. res

3. (3) 处应填 ()。

- A. (dir[i]+sum)%2!=0 B. dir[i]%2!=0
C. dir[i]+sum!=0 D. (dir[i]+sum)%2==0

4. (4) 处应填 () 。

- A. sum B. res C. f[k] D. f[n]

5. (5) 处应填 ()。

- A. m==-1 B. M>m C. m>=0&&M>m D. M>=m