2019年厦门大学铃盛杯C语言积分赛第一周周赛题解

A题：人工智障

出题人：691

难度评价：**Hard**

解法:
一道非常恶心的模拟题。细节非常多。
这题的灵感来自于今年春季的PTA天梯赛。（当然，细节差别非常多，不可能复制黏贴或简单修改通过）
具体细节看代码吧。建议采用工程化码风来减少编程复杂度

一组hack数据：

```
10
?
I
me.
I love myself!
I? I ! you. you , me. me ? your.
you???
your???!!!
im
.
```

std1:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#define maxn 5005
const char cword[3][4]={"I","me","you"};
const int cwordlen[3]={1,2,3};
const int dwordlen[3]={3,3,1};
const char dword[3][4]={"you","you","I"};
void change_punct(char * str,int len)
{
    for (int i=0;i<len;++i)
        if (str[i]=='?')
            str[i]='!';
}
int erase_blank(char * str,int len)
{
    char temp[maxn];
    int i=0,j=0;
    while (i<len&&str[i]==' ') ++i;
```

```
    for (;i<len;++i)
    {
        if (ispunct(str[i+1])&&str[i]==' ')continue;
        if (str[i]==' '&&str[i+1]==' ') continue;
        temp[j++]=str[i];
    }
    for (i=j-1;i>=0&&temp[i]==' ';--i);
    temp[i+1]='\0';
    strcpy(str,temp);
    return i+1;
}
int change_word(char *str,int len)
{
    char temp[maxn];
    int i=0,newlen=0;
    while (i<len&&(str[i]==' '||ispunct(str[i]))) ++i;
    memcpy(temp,str,sizeof(char)*i);
    newlen+=i;
    for (;i<len;)
    {
        int j=i,k;
        while (j<len&&(str[j]!=' '&&!ispunct(str[j]))) ++j;
        for (k=0;k<3;++k)
            if (j-
i==cwordlen[k]&&memcmp(str+i,cword[k],cwordlen[k]*sizeof(char))==0)
            {
                strcpy(temp+newlen,dword[k]);
                newlen+=dwordlen[k];
                break;
            }
        if (k==3)
        {
            memcpy(temp+newlen,str+i,(j-i)*sizeof(char));
            newlen+=j-i;
        }
        i=j;
        while (i<len&&(str[i]==' '||ispunct(str[i]))) ++i;
        memcpy(temp+newlen,str+j,(i-j)*sizeof(char));
        newlen+=i-j;
    }
    temp[newlen]='\0';
    strcpy(str,temp);
    return newlen;
}
int main(void)
{
    int t;
    char str[maxn];
    scanf("%d ",&t);
    for (int i=0;i<t;++i)
    {
        fgets(str,maxn,stdin);
        int len=strlen(str);
        str[--len]='\0';
```

```
        change_punct(str,len);
        len=erase_blank(str,len);
        change_word(str,len);
        puts(str);
    }
    return 0;
}
```

std2:

```
#include <stdio.h>
#include <string.h>
char bin[1010][1010];
bool read(int tot)
{
        char c = getchar();
        while (c == ' ')
        {
                c = getchar();
        }
        int cnt = -1;
        while (c != ' '&&c != '\n'&&c!=EOF)
        {
                if (c != '\r')
        {
            if(c=='?')c='!';
            bin[tot][++cnt] = c;
        }
                c = getchar();
        }
        bin[tot][++cnt] = '\0';
        if (c == '\n'||c==EOF)return 0;
        return 1;
}
bool cmp(char *a,const char* b)
{
    int tot=0;
    while(b[tot]!='\0')
    {
        if(a[tot]!=b[tot])return 0;
        ++tot;
    }
    if(a[tot]=='\0'||a[tot]=='?'||a[tot]=='.'||a[tot]=='!'||a[tot]==',')return 1;
    else return 0;
}
void cpy(char *a,const char* b)
{
    char q[1010];int qi=-1;
    int tot=0;
    while(a[tot]!='\0')
    {
        if(a[tot]=='?'||a[tot]=='.'||a[tot]=='!'||a[tot]==',')q[++qi]=a[tot];
```

```
            ++tot;
        }
        tot=0;
        while(b[tot]!='\0')
        {
            a[tot]=b[tot];
            ++tot;
        }
        for(int i=0;i<=qi;i++)a[tot++]=q[qi];
        a[tot]='\0';
    }
    int main()
    {
            freopen("testdata.in", "r", stdin);
            freopen("testdata.out", "w", stdout);
            int T;
            scanf("%d", &T); read(0);
            while (T--)
            {

                    int tot = 1;
                    while (read(tot))++tot;
                    for (int i = 1; i <= tot; i++)
                    {
                            if (cmp(bin[i], "I"))cpy(bin[i], "you");
                            else if (cmp(bin[i], "me"))cpy(bin[i], "you");
                            else if (cmp(bin[i], "you"))cpy(bin[i], "I");
                    }
                    for (int i = 1; i <= tot; i++)
                    {
                            if (bin[i][0] == '\0')continue;
                            if (i > 1 && bin[i][0] != '?'&&bin[i][0] != '!'&&bin[i][0]
    != ','&&bin[i][0] != '.')printf(" ");
                            int j = 0;
                            while (bin[i][j] != '\0')putchar(bin[i][j++]);
                    }
                    printf("\n");
            }
            return 0;
    }
```

B. 三星拉克丝

出题人:lrl

难度评价：**Very Hard**

解法：

一道概率题，考察选手的数学能力和代码优化能力。代码量很小，但是难度最高。

枚举抽牌数量i，计算抽牌数为i时能够抽到三星拉克丝的概率pi，答案为$\sum_{i=9}^{n} i * pi$

其中，$pi = \frac{C_{m-8}^{1}}{C_{n-i+1}^{1}} * \frac{C_{m}^{8} C_{n-m}^{i-9}}{C_{n}^{i-1}}$

std:

```
#include<stdio.h>
double N,M;
int main()
{
    int T;
    scanf("%d", &T);
    while(T--)
    {
        scanf("%lf%lf", &N, &M);
        double ans=0, pre=9;
        for(int i=0;i<=8;i++)
        {
            pre*=M-i;
            pre/=N-i;
        }
        ans = pre;
        for(int i=10;N-M-i+10>0;i++)
        {
            pre *= i;
            pre /= i-9;
            pre /= N-i+1;
            pre *= N-M-i+10;
            ans += pre;
        }
        printf("%.3lf\n",ans);
    }
    return 0;
}
```

C. 小吴的纸带

出题人: xyc

难度评价：**Easy**

解法：见注释

std：

```
// 本题为简单的模拟题，只要将题面翻译为 C 语言即可，并注意不要缺漏题意
// 题面与代码对应关系见代码注释

#include <stdio.h>

int main() {
```

```
        char str[5001];
        int sum, index, offset;
        scanf("%s", str);
        scanf("%d", &sum);
        scanf("%d %d", &index, &offset);// 对应“阅读器首先载入第一个操作”
        int numofins = 0;
        for (int curri = 0; str[curri] != 0; ) {
        // 初始化语句对应“开始读取字符串”
        // 条件语句对应“当且仅当阅读器越过结尾的最后一个字符时，阅读器停止工作”
                printf("%c", str[curri]);// 对应“读到一个字母时，首先输出它”
                if (curri != index || numofins == sum) {// 对应“如果它的位置和当前操
作的 a 相等”
        // 及“如果所有操作都已执行完毕”
                        ++curri;// 对应“将纸带后移1位”
                }
                else {// 对应“如果当前位置与 a 不相等”
                        curri += offset;// 对应“将纸带后移 b 位”
                        if (numofins != sum) {// 对应“（如果有的话）”
                                scanf("%d %d", &index, &offset);// 对应“载入下一个
操作”
                                ++numofins;
                        }
                }
        }
        return 0;
}
```

D. 不许喷漆

出题人：691

难度评价：**Easy**

解法：

我们不妨将整个纸带划分成k块颜色相同的连续段。(具体一段有多长显然无关紧要)

当k为奇数时，显然两端颜色相同。这时候很容易想到的策略就是从外到里一层一层向内铺(参考样例中的ababa，每次铺$[1,k],[2,k-1]...[k/2+1]$,一共铺$k/2+1$次.

当k为偶数时，两端颜色不同.

一个比较容易理解的操作是将一端的颜色补到与另一端相同，比如$bbababaa$补成$abbababaa$，显然它们两个是等价的，因为铺最外层的a可以兼顾两头.

也可以很容易发现这里的答案也是$k/2+1$（k为偶数）

那么最后答案就是$k/2+1$.

std:

```
std:#include<stdio.h>
#define MAXN 1000010
#define INF 691691691
char bin[MAXN];
int main()
{
    int T;
    scanf("%d",&T);
    while(T--)
    {
        int N;
        scanf("%d",&N);
        scanf("%s",&bin);
        int cnt=0;
        char now=0;
        for(int i=0;i<N;i++)
        {
            if(now!=bin[i])
            {
                ++cnt;
                now=bin[i];
            }
        }
        printf("%d\n",cnt/2+1);
    }
    return 0;
}
```

E. 数字合成游戏

出题人：lzz

难度评价：**Normal**

解法：

注意到三个事实：

1. 任意加括号，相当于我们总是能让相邻的两个数字进行运算。
2. 每次进行一次运算（加或乘），数字就少一个，当只剩一个数字时就是最终答案

因此我们很容易设计一个递归搜索过程，设深度为6，每次递归深度减1，当深度为1时说明搜索结束，用一个数组记录答案。

每次递归，枚举两个相邻的数字将他们合并（也就是加法或乘法），并将新的数组作为参数传递给递归函数。

可计算出时间复杂度约为$5! * 2^5$，可以很快速的通过全部测试数据。

std:

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#define maxrange 531442
bool cnt[maxrange];
int ans=0;
void dfs(int depth,int *ar)
{
    if (depth==1)
    {
        if (cnt[ar[0]]==1);
        else
        {
            cnt[ar[0]]=1;
            ++ans;
        }
        return;
    }
    int ar2[6];
    for (int i=1;i<depth;++i)
    {
        for (int j=0;j<i-1;++j)
            ar2[j]=ar[j];
        for (int j=i;j<depth-1;++j)
            ar2[j]=ar[j+1];
        ar2[i-1]=ar[i]+ar[i-1];
        dfs(depth-1,ar2);
        ar2[i-1]=ar[i]*ar[i-1];
        dfs(depth-1,ar2);
    }
}
int main(void)
{
    int ar[6];
    for (int i=0;i<6;++i)
        scanf("%d",&ar[i]);
    dfs(6,ar);
    printf("%d",ans);
    return 0;
}
```