天津大学

程序设计综合实践课程报告

字符串和数学实验

学生姓名 陈秋澄

学院名称 智能与计算学部

专　　业 大类

学　　号 **3022244290**

1. 字符串

1.1 题目分析

使用指针和具有查找子串第一次出现位置功能的函数 strstr，进行求解。

1.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int T;
    cin>>T;
    while(T--){
        char a[1001];
        char b[1001];
        cin>>a;
        cin>>b;
        char *p=strstr(a,b);     //运用函数，直接查找
        if(p!=NULL)
            cout<<p-a<<endl;     //输出位数
        else
            cout<<-1<<endl;
    }
}
```

2. Oulipo

2.1 题目分析

利用 KMP 算法，但是每次找到一个后，ans++， 但不要立即跳出来，而是让 j= next[j]。

2.2 题目代码

```
#include <iostream>
#include <stdio.h>
#include <string>
#include <string.h>
using namespace std;
const int maxn = 1000005;
int Next[maxn], ans;          //出现次数
void GetNext(string p, int * Next)
{
    int p_len = p.length();  //字符串长度
    int i = 0;
    int j = -1;
    Next[0] = -1;
    while(i < p_len)
    {
        if(j == -1 || p[i] == p[j])
        {
            i++;
            j++;
            Next[i] = j;
        }
        else
        j = Next[j];
    }
}

int KMP(string s, string p, int * Next)
{
    GetNext(p, Next);
    int s_len = s.length();
```

```
    int p_len = p.length();
    int i = 0;
    int j = 0;
    while(i < s_len)          //判断是否有相同字符出现
    {
        if(j == p_len)
        {
            ans++;
            j = Next[j];
        }
        if(j == -1 || s[i] == p[j])
        {
            i++;
            j++;
        }
        else
        j = Next[j];
    }
    if(i == s_len && j == p_len)//字符串相等
    ans++;
    return ans;
}
int main()
{
    int t;
    scanf("%d", &t);
    while(t--)
    {
        string s1, s2;
        cin>>s2>>s1;
        ans = 0;
        KMP(s1, s2, Next);
        printf("%d\n", ans);
    }
    return 0;
}
```

3. 本质不同的子串

3.1 题目分析

使用 map 函数将字符串和数字联系起来；
使用较为简单的方法，将字符串复制一遍，这样可将字符环转为熟悉的字符串。
“加长”字符串的长度不大于原字符串长度的子串个数即为所求。

3.2 题目代码

```
#include <iostream>
#include <algorithm>
#include <map>
#include <hash_map>//需要的全部头文件
using namespace std;
int main()
{
	int n;
	cin >> n;
	for (int i=n;n>0;n--)
	{
		int num = 0;
		int len;
		string s0;
		cin >> s0;
		len = s0.size();		//长度
		string str = s0;
		map<string, int> map0;
		str=s0+s0;// 将字符串复制一遍，这样可将字符环转为熟悉的字符串。
//“加长”字符//串的长度不大于原字符串长度的子串个数即为所求。
		for (int i = 0; i < len; i++)
		{
			for (int j = 1; j <= len; j++)
			{
				string str2 = str.substr(i, j);
				if (map0[str2] == 0)
				{
					num++;
					map0[str2]++;
				}
```

```
                }
            }
            cout << num << endl; //输出由 map 函数匹配出的结果
        }
}
```

4. 最小公倍数

4.1 题目分析

使用枚举法，取两数中较小者，从它开始至两数乘积，遍历。

4.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    long long a,b;
    while(cin>>a&&cin>>b){
        int c=min(a,b);
        for(long long i=c;i<=a*b;i++){
            if(i%a==0&&i%b==0){
                cout<<i<<endl;
                break;
            }
        }
    }
}
```

5. 素数求和

5.1 题目分析

首先将 2-n 的数全部标记为素数：

由小到大遍历每一个标记为素数的数字，将该素数的倍数全部置为标记为非素数，最后求和。

5.2 题目代码

```
#include<bits/stdc++.h>
using namespace std;
const int maxn = 2e7 + 50;
int flag[maxn],n;
long long ans;
int main(){
    std::ios::sync_with_stdio(0);
    while(cin>>n){
        for(int i = 2;i<= n;i++){
            if(flag[i] == 0){                    //如果为素数
                for(int j = i*2;j<= n;j+= i){    //标记该素数的倍数
                    flag[j] = 1;
                }
            }
        }
        for(int i = 2;i<= n;i++){                //求和
            if(flag[i] == 0){
                ans += i;
            }
        }
        cout<<ans<<endl;
    }
    //默认 0 为素数，1 为非素数
        return 0;
}
```

6. 人见人爱 A^B

6.1 题目分析

题目要求只要求最后三位，如果直接将结果算出来之后再%1000 是不行的，因为这个结果可能会非常非常大，所以可以每次求出 A*A 的最后三位，之后再乘以 A。

6.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int fun(int a,int b){
    int x=a;
    for(int i=1;i<b;i++){
        x*=a;
        if(x>999)
            x%=1000;            //求出 A*A 的最后三位
    }
    return x;
}

int main(){
    int a,b;
    while(cin>>a>>b&&(a!=0||b!=0)){
        cout<<fun(a,b)<<endl;

    }
}
```

7. XORinacci

7.1 题目分析

发现进行 3 次操作就会回到原来的数，所以只需将 n%3，再讨论一下，进行^运算即可。

7.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int T;
    cin>>T;
    while(T--){
        int a=0,b=0,c=0;
        int f[3];
        cin>>a>>b>>c;
        c%=3;
        f[0]=a;
        f[1]=b;
        f[2]=a^b;              //3 为一个规律循环
        cout<<f[c]<<endl;
    }
}
```

8. 不同的 n/i

8.1 题目分析

根据规律，如下表所示：

1	**2=1*2**	**3**	**4**	**5**	**6=2*3**	**7**	**8**	**9**	**10**
1	**2**	**2**	**3**	**3**	**4**	**4**	**4**	**5**	**5**

以平方根为单位，两个平方根内发生变换的数字为前一个平方根乘以下一个平方根。

8.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int T;
    cin>>T;
    while(T--){
        long long a=0;
        cin>>a;
        long long b=sqrt(a);
        if(a>=b*(b+1)){        //判断是否处在对应区间
            cout<<2*b<<endl;
        }
        else cout<<2*b-1<<endl;
    }
}
```