**实验报告**

**课程名称：算法设计与分析**

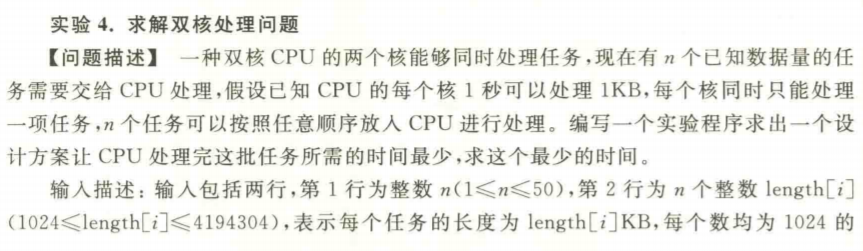
**专业班级：大数据182**

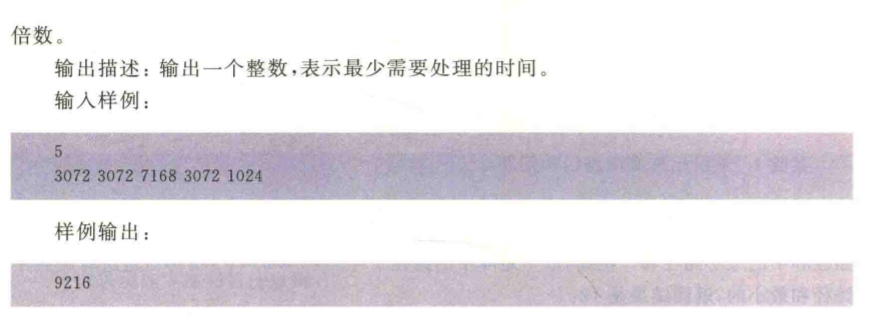
**学号姓名：3180439031陈佳婧**

**实验日期：2020年12月7日**

**第八章上机实验题4： 求解双核处理问题**

1. **实验目标和要求：**





**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

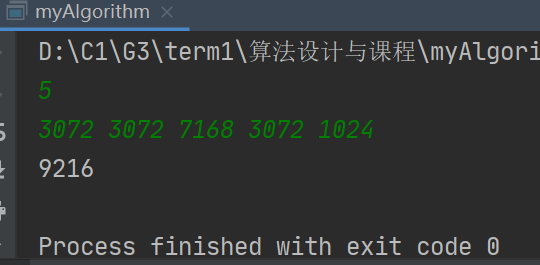
**0/1背包dp。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
#include <algorithm>  
using namespace std;  
const int N = 5000;  
int dp[N];//N代表重量  
  
int main() {  
 int n;  
 int work[55];  
 cin>>n;  
 int sum=0;  
 for(int i=0;i<n;i++){  
 cin>>work[i];  
 work[i]/=1024;  
 sum+=work[i];  
 }  
 for(int i=0;i<n;i++){  
 for(int j=sum/2;j>=work[i];j--){  
 dp[j]=max(dp[j],dp[j-work[i]]+work[i]);//选或不选，不断逼近sum/2  
 }  
 }  
 cout<<max(dp[sum/2],sum-dp[sum/2])\*1024<<endl;  
 return 0;  
}

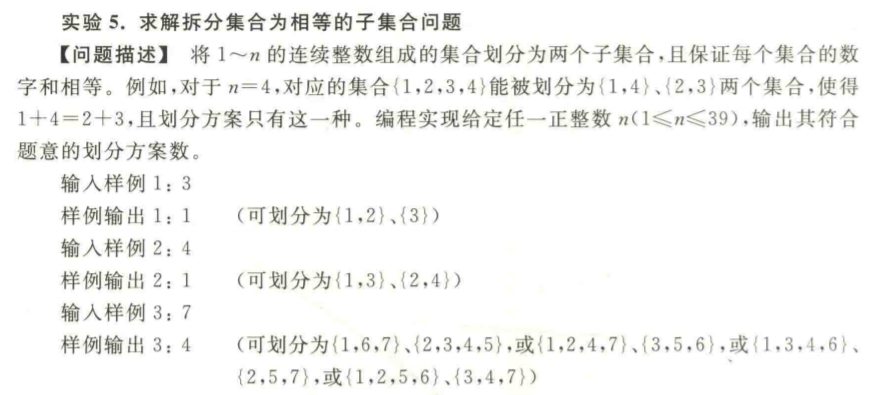
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第八章上机实验题5： 求解拆分集合为相等的子集合问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

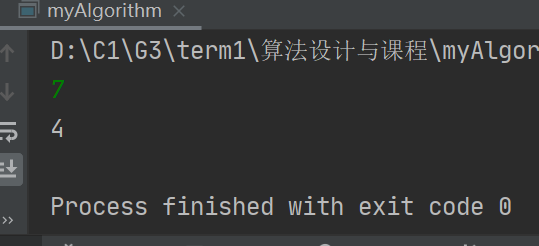
dp.

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
#include <algorithm>  
using namespace std;  
const int N = 50;  
  
  
int main() {int dp[N]={0};//N代表重量  
 int n;  
 int work[55];  
 cin>>n;  
 int sum=(1+n)\*n/2;  
 if(sum%2)cout<<0<<endl;  
 else{  
 for(int i=0;i<n;i++){  
 work[i]=i+1;  
 }  
 dp[0]=1;  
 for(int i=0;i<n;i++){  
 for(int j=sum/2;j>=work[i];j--){  
 dp[j]=dp[j-work[i]]+dp[j];//选或不选，不断逼近sum/2  
 }  
 }  
 cout<<dp[sum/2]/2<<endl;  
 }  
 return 0;  
}

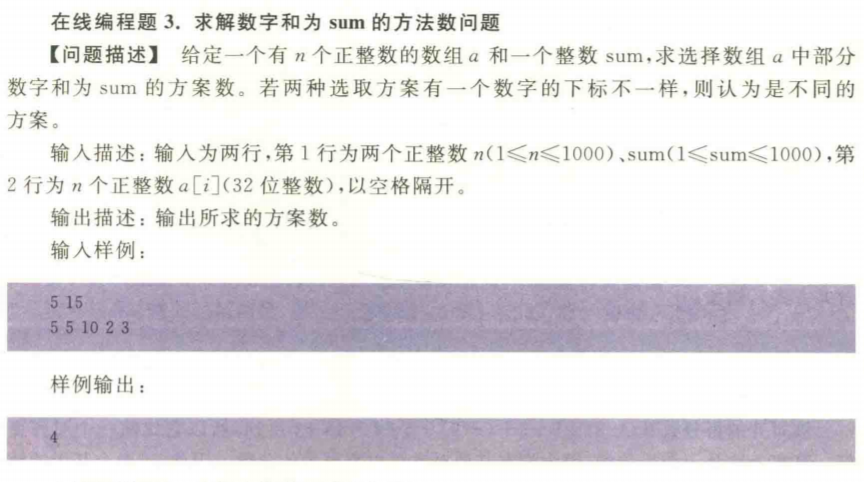
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第八章在线编程题3： 求解数字和为sum的方法数问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

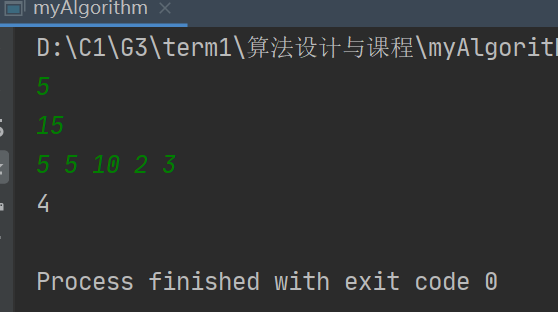
**dp**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
#include <algorithm>  
  
using namespace std;  
const int N = 50;  
  
int main() {  
 int dp[N] = {0};//N代表重量  
 int n, sum;  
 int work[55];  
 cin >> n >> sum;  
 for (int i = 0; i < n; i++) {  
 cin >> work[i];  
 }  
 dp[0]=1;  
 for (int i = 0; i < n; i++) {  
 for (int j = sum; j >= work[i]; j--) {  
 dp[j] = dp[j - work[i]] + dp[j];  
 }  
 }  
 cout << dp[sum] << endl;  
  
 return 0;  
}

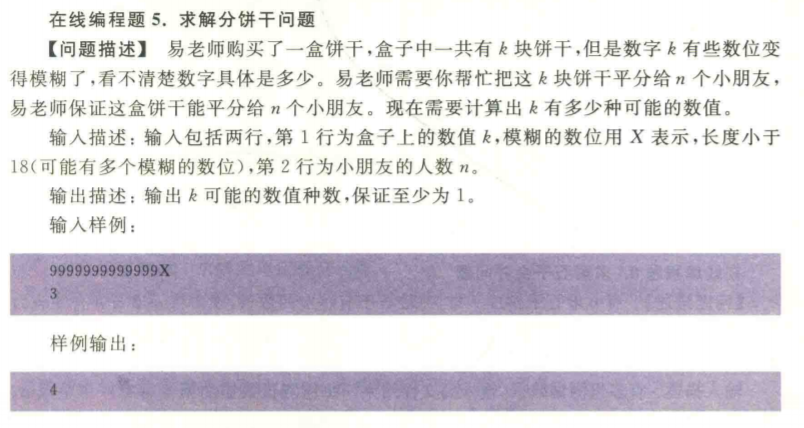
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第八章在线编程题2： 求解分饼干问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

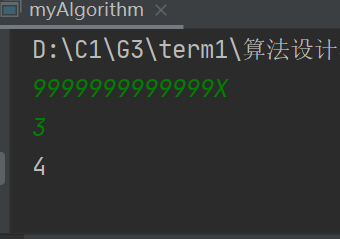
**dp。**

**4. 程序流程：**

**5. 完整程序代码：**

#include<iostream>  
using namespace std;  
  
string str;  
int n;  
const int maxn = 100;  
int dp[maxn][maxn]; //dp[i][j]前i位字符串%n等于j的情况数  
int len;  
  
/\*  
9999999999999X  
3  
\*/  
int DP() {  
 //边界，第一位  
 if (str[0] == 'X') {  
 for (int j = 0; j <= n - 1; j++) //枚举余数  
 {  
 for (int i = 0; i <= 9; i++) {  
 if (i % n == j)  
 dp[1][j]++;  
 }  
 }  
 } else {  
 int x = str[0] - '0';  
 int j = x % n;  
 dp[1][j]++;  
 }  
 //后面的位数  
 for (int i = 2; i <= len; i++) {  
 for (int j = 0; j <= n - 1; j++) {  
 if (str[i - 1] == 'X') {  
 for (int k = 0; k <= 9; k++) {  
 //这里要理解状态转移  
 int newj = (10 \* j + k) % n;  
 dp[i][newj] += dp[i - 1][j];  
 }  
 } else {  
 int newj = (10 \* j + str[i - 1] - '0') % n;  
 dp[i][newj] += dp[i - 1][j];  
 }  
 }  
 }  
 return dp[len][0];  
}  
  
int main() {  
 cin >> str;  
 len = str.length();  
 cin >> n;  
 cout << DP() << endl;  
 return 0;  
}

**6. 运行结果展示：**



**7. 实验体会（可选）：**