****



**程序设计综合实践课程报告**

**基础算法2实验**

**学生姓名**

**学院名称 智能与计算学部**

**专 业 工科试验班(智能与计算类)**

**学 号**

# 1. 找零

## 1.1题目分析

**利用贪心法，总是先用最大金额的纸币，当最大金额纸币超过当前剩余金额再依次向下，直到剩余金额减为0。**

## 1.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;    int main() {      int n;      cin >> n;      while (n--) {          int total, i;          cin >> total;          int money[7] = {1, 2, 5, 10, 20, 50, 100};          int num[7] = {0};          for (i = 0; i < 7; i++)              cin >> num[i];            int counts = 0;          for (i = 6; i > 0; i--) {              int temp = total / money[i];              if (temp <= num[i]) {                  counts += temp;                  total = total % money[i];              } else {                  counts += num[i];                  total -= (num[i] \* money[i]);              }          }          if (total <= num[0])              counts += total;          else              counts = -1;          cout << counts << endl;      }      return 0;  } |
|  |

# 2. Rightmost Digit

## 2.1题目分析

**通过不断的幂取模运算求得最后一位。**

## 2.2 题目代码（带注释）

|  |
| --- |
| #include<iostream>  using namespace std;    int main() {      int T;      cin >> T;      while (T--) {          long long int n;          cin >> n;          long long int sum = 1;          long long int temp = n;          for (; n != 0; n /= 2) {              if (n % 2 == 1)                  sum \*= temp;              temp \*= temp;                sum %= 10;              temp %= 10;          }          cout << sum << endl;      }      return 0;  } |
|  |

# 3. Monthly Expense

## 3.1题目分析

**二分法，所求答案ans在[low,high]范围内二分。**

**low是最高的单月花费；high是各个月所有花费之和，mid是当前的ans。**

**遍历各个月份，将花费总和不超过ans的月份归入一个组，统计在当前ans情况下能分成几个组。**

**如果组数少于M，说明ans偏大，范围变成前二分之一；**

**否则说明ans偏小，范围变成后二分之一。**

## 3.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  using namespace std;  int N, M;  int money[101];    bool judge(int max, int money[]) {      /\* 检测以 max 为标准是否能分成 M 组 \*/      int sum = 0;      int group = 1;      for (int i = 1; i <= N; i++) {          if (sum + money[i] <= max)              sum += money[i];          else {              sum = money[i];              group++;          }      }      if (group > M)          return false;      else          return true;  }    int main() {      cin >> N >> M;      int high = 0, low = 0;      for (int i = 1; i <= N; i++) {          cin >> money[i];          high += money[i]; // total money          if (low < money[i])              low = money[i]; // maximum money      }      int mid, ans;      mid = (low + high) / 2;      while (low <= high) {          if (!judge(mid, money))              low = mid + 1;          else {              ans = mid;              high = mid - 1;          }          mid = (low + high) / 2;      }      cout << ans << endl;      return 0;  } |
|  |

# 4. 三角形个数

## 4.1题目分析

**首先对各条边进行排序，然后用三重循环利用三角形三边的条件去限定。**

## 4.2 题目代码（带注释）

|  |
| --- |
| #include<iostream>  #include<algorithm>  #include<stdio.h>  using namespace std;    int main()  {      int n, a[1000], ans = 0;      cin >> n;      for (int i = 0; i < n; i++)          cin >> a[i];      sort(a, a + n);      for (int i = 0; i < n - 2; i++) {          for (int j = i + 1; j < n - 1; j++) {              for (int k = j + 1; k < n; k++) {                  if (a[i] + a[j] <= a[k])                      break;                  else                      ans++;              }          }      }      cout << ans << endl;      return 0;  } |
|  |

# 5. Greed

## 5.1题目分析

**是一个将剩余的可乐是否可以倒入两个杯子的问题，先将杯子最大容量和次大容量求出，如果剩余的可乐体积小于这两个杯子的容量，则可以，即“YES”，否则“NO”。**

## 5.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <algorithm>  using namespace std;  int main()  {      long long n, sum;      int a[100000], b[100000];      cin >> n;      sum = 0;      for (int i = 0; i < n; i++) {          cin >> a[i];          sum += a[i];      }      for (int i = 0; i < n; i++)          cin >> b[i];      sort(b, b + n);      if (sum > b[n - 2] + b[n - 1])          cout << "NO" << endl;      else          cout << "YES" << endl;      return 0;  } |
|  |

# 6. 珠心算测验

## 6.1题目分析

**其中有多少个数，恰好等于集合中另外两个（不同的）数之和，等于两个数之和的那个数是最后的，前面的两个数是相加的，那么在开始的时候就可以用sort进行排序。**

## 6.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <algorithm>  #include <stdio.h>  using namespace std;    int main()  {      long long n, a[1010], ans = 0, flag;      cin >> n;      for (int i = 0; i < n; i++)          cin >> a[i];      sort(a, a + n);      for (int k = 2; k < n; k++) {          flag = 0;          for (int i = 0; i < k - 1; i++) {              for (int j = i + 1; j < k; j++)                  if (a[k] == a[i] + a[j]) {                      flag = 1;                      ans++;                      break;                  }              if (flag)                  break;          }      }      cout << ans << endl;        return 0;  } |
|  |

# 7. Long Number

## 7.1题目分析

**从头开始找，找到第一个需要替换的，从那个位置开始替换。**

## 7.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <algorithm>  #include <stdio.h>  using namespace std;  int main()  {      int n;      string s;      char a[9];      cin >> n;      cin >> s;      for (int i = 0; i < 9; i++)          cin >> a[i];      for (int i = 0; i < 9; i++)          if (a[i] - '0' < i)              a[i] = i + '1';      for (int i = 0; s[i]; i++)          s[i] = a[s[i] - '1'];      cout << s << endl;      return 0;  } |
|  |