

Yêu cầu chung:

- Đăng nhập vào hệ thống thi bằng email SV trường cấp.
- Dữ liệu vào luôn luôn đúng đắn, sinh viên không cần kiểm tra.
- SV làm xong bài nào, nộp bài đó ngay (copy code dán vào trình chấm) để xem kết quả, mỗi bài SV được nộp nhiều lần, bài nào cao điểm nhất sẽ được tính.
- Tất cả dữ liệu vào được nhập từ thiết bị nhập chuẩn (nhập từ bàn phím trên màn hình console) và tất cả dữ liệu ra đều xuất ra thiết bị xuất chuẩn (màn hình console). Thời gian chạy chương trình cho mỗi bài là 1 giây.

ĐỀ BÀI

Bài 1: CUAHANG (2.5 điểm)

Nhân dịp mừng kỷ niệm ngày thành lập, một cửa hàng đồng giá có hình thức giảm giá và khuyến mãi như sau:

- Mua dưới 5 món: không giảm giá.
- Mua từ 5 – 10 món: giảm giá 5%/món.
- Mua từ 11 – 20 món: giảm giá 10%/món.
- Mua từ 21 món trở lên: giảm giá 15%/món.

Nếu khách mua vào ngày thứ ba hoặc thứ tư hàng tuần, khách hàng sẽ được nhận một phiếu quà tặng cho mỗi 10 món hàng đã mua.

Yêu cầu: Một khách hàng mua n món hàng ($n < 50$) vào ngày thứ t ($t=1$ biểu thị ngày Chủ nhật), giá mỗi món hàng trước khi giảm giá là p ($p < 10^8$) đồng. Tính số tiền mà khách hàng phải trả và số phiếu quà tặng được nhận.

Dữ liệu vào: Lần lượt là ba số nguyên dương n , t và p (mỗi số cách nhau ít nhất một khoảng trắng).

Dữ liệu ra: Hai số nguyên lần lượt thể hiện số tiền phải trả (làm tròn đến phần nguyên) và số phiếu quà tặng khách được nhận (mỗi số cách nhau ít nhất một khoảng trắng).

Ví dụ:

Dữ liệu vào	
7	5 2000000
Dữ liệu ra	
13300000	0

Dữ liệu vào	
24	3 1850000
Dữ liệu ra	
37740000	2

Bài 2: TROCHOI (2.5 điểm)

Cho A là một mảng các số nguyên có n phần tử đại diện n người chơi một trò chơi. Trong đó, phần tử cuối cùng của mảng đại diện cho **người chủ trò chơi**, n-1 phần tử đầu tiên là của **những người tham gia trò chơi**. Số điểm của người chơi thứ i ($0 \leq i < n-1$) được tính bằng tổng các chữ số của phần tử thứ i.

Kết quả của trò chơi được tính bằng cách so sánh lần lượt số điểm giữa người chủ trò chơi và **từng** người tham gia trò chơi: nếu người chủ trò chơi có số điểm lớn hơn người chơi thứ i, kết quả tăng thêm 1. Ngược lại, nếu chủ trò chơi có số điểm nhỏ hơn hoặc bằng, sẽ trừ một điểm vào kết quả (kết quả ban đầu bằng 0).

Yêu cầu: Hãy tính kết quả sau một trò chơi.

Dữ liệu vào có cấu trúc như sau:

- Dòng đầu tiên là một số nguyên dương duy nhất n ($2 \leq n \leq 150$) biểu diễn số phần tử của mảng A.
- Dòng tiếp theo là n số nguyên lớn hơn -1 và nhỏ hơn 10000 (mỗi số cách nhau ít nhất một khoảng trắng) lần lượt là n phần tử của mảng A.

Dữ liệu ra: Một số nguyên duy nhất thể hiện kết quả.

Ví dụ:

Dữ liệu vào
4
123 270 50 44
Dữ liệu ra
1

Dữ liệu vào
5
11 2731 32 225 5
Dữ liệu ra
-2

Giải thích cho ví dụ 1:

- Người chủ trò chơi ($A[3]$) có số điểm $4+4=8$.

- Kết quả:

+ Người chơi thứ 1 ($A[0]$) có số điểm $1+2+3=6 < 8 \rightarrow$ kết quả +1

+ Người chơi thứ 2 ($A[1]$) có số điểm $2+7+0=9 > 8 \rightarrow$ kết quả -1

+ Người chơi thứ 3 ($A[2]$) có số điểm là $5 < 8 \rightarrow$ kết quả +1

Bài 3: DMTU (2.5 điểm)

Cho một chuỗi ký tự S chỉ gồm các ký tự chữ cái ('a' đến 'z', 'A' đến 'Z') và ký tự trắng.

Yêu cầu: Đếm số từ có chứa ít nhất 2 nguyên âm đứng cạnh nhau, với quy ước:

- Từ là một dãy liên tiếp các ký tự khác trắng và được ngăn cách với nhau bởi khoảng trắng.
- Nguyên âm gồm các ký tự A, E, I, O, U, Y, a, e, i, o, u và y.

Dữ liệu vào: Một chuỗi ký tự S (chiều dài tối đa 200 ký tự).

Dữ liệu ra: Một số nguyên duy nhất cho biết giá trị đếm được.

Ví dụ:

Dữ liệu vào
Nguyen Thi HOA mal
Dữ liệu ra
3

Dữ liệu vào
I am a student
Dữ liệu ra
0

Bài 4: TBCMIN (2.5 điểm)

Cho ma trận A kích thước $M \times N$ ($0 < M, N < 100$) chứa các số nguyên dương nhỏ hơn 100000.

Yêu cầu: Tính trung bình cộng của các phần tử nhỏ nhất trên mỗi dòng (làm tròn đến phần nguyên theo nguyên tắc phần thập phân < 0.5 thì làm tròn xuống, ngược lại làm tròn lên).

Dữ liệu vào: Có cấu trúc như sau:

- Dòng đầu tiên là hai số nguyên dương M, N biểu diễn kích thước của ma trận A (M dòng, N cột).
- M dòng tiếp theo, mỗi dòng là N số nguyên (mỗi số cách nhau ít nhất một khoảng trắng) lần lượt là N phần tử của từng dòng tương ứng của ma trận.

Dữ liệu ra: Một số nguyên duy nhất cho biết giá trị tính được.

Ví dụ:

Dữ liệu vào			
3	4		
3	1	5	6
4	6	2	3
8	9	7	6

Dữ liệu ra
3

-HẾT-

Ghi chú:

- Giám thị coi thi không giải thích đề thi.

Chuẩn đầu ra của học phần	Nội dung kiểm tra
[G2.2]: Ứng dụng các kiểu dữ liệu cơ bản, biến, hằng, các phép toán và cấu trúc điều khiển của ngôn ngữ C/C++ trong lập trình giải các bài toán đơn giản. [G2.3]: Sử dụng được kiểu dữ liệu trừu tượng, mảng, chuỗi, hàm, các lệnh nhập/xuất có định dạng trên chương trình viết bằng C/C++.	Bài 1, 2, 3, 4
[G4.1]: Phát hiện và xử lý vấn đề trong giải bài toán lập trình. [G4.2]: Thể hiện thái độ làm việc chăm chỉ, chú ý đến chi tiết.	Bài 1, 2, 3, 4

Ngày 10 tháng 11 năm 2023

K/T Trưởng bộ môn


Trần Công Tư

1 // Đây là đáp án mẫu. SV có thể làm theo cách khác. Bất cứ
giải pháp nào chạy được và đúng các testcase là có điểm.

2
3 // Bai 1

4 #include <iostream>

5 using namespace std;

6
7 int main() {

8 // Declaration

9 long long n, t, p;

10 long long soTien, soPhieu;

11
12 // Get input

13 cin >> n >> t >> p;

14
15 // Processing

16 if (n < 5) {

17 soTien = n * p;

18 } else if (n <= 10) {

19 soTien = n * p * 0.95;

20 } else if (n <= 20) {

21 soTien = n * p * 0.9;

22 } else {

23 soTien = n * p * 0.85;

24 }

25
26 if (t == 3 || t == 4) {

27 soPhieu = n / 10;

28 }

29
30 // Print output

31 cout << soTien << " " << soPhieu;

32
33 return 0;

34 }

35
36
37 // Bai 2

38 #include <iostream>

39 using namespace std;

40
41 const int N = 150;

42
43 int soDiem(int x) {

44 int diem = 0;

45 while (x > 0) {

46 diem += x % 10;

47 x /= 10;

48 }

```

49         return diem;
50     }
51
52
53     int main() {
54         // Declaration
55         int n, i;
56         int a[N];
57         int kq = 0;
58         int diemNhaCai;
59
60         // Get input
61         cin >> n;
62         for (i = 0; i < n; i++) {
63             cin >> a[i];
64         }
65
66         // Processing
67         diemNhaCai = soDiem(a[n - 1]);
68         for(i = 0; i < n - 1; i++) {
69             if (soDiem(a[i]) < diemNhaCai) {
70                 kq += 1;
71             } else {
72                 kq -= 1;
73             }
74         }
75
76         // Print output
77         cout << kq;
78
79         return 0;
80     }
81
82
83
84     // Bai 3
85     #include <iostream>
86     #include <string>
87     using namespace std;
88
89     bool kiemTra2NALienTiep(string s) {
90         string na = "AEIOUYaeiouy";
91
92         for(int i = 0; i < s.size() - 1; i++) {
93             if (na.find(s[i]) != -1 && na.find(s[i+1]) != -1) {
94                 return true;
95             }
96         }
97

```

```

98     return false;
99 }
100
101 int main() {
102     // Declaration
103     string s;
104     int d;
105
106     // Get input and processing
107     d = 0;
108     while (cin >> s) {
109         if (kiemTra2NALienTiep(s)) {
110             d++;
111         }
112     }
113
114     // Print output
115     cout << d;
116
117     return 0;
118 }
119
120
121 // Bai 4
122 #include <iostream>
123 #include <cmath>
124 using namespace std;
125
126 const int M = 100;
127 const int N = 100;
128
129 int minDong(int a[], int n) {
130     int m = a[0];
131     for(int i = 1; i < n; i++) {
132         if (a[i] < m) {
133             m = a[i];
134         }
135     }
136     return m;
137 }
138
139 int main() {
140     // Declaration
141     int m, n, i, j, s;
142     int a[M][N];
143
144     // Get input
145     cin >> m >> n;
146     for (i = 0; i < m; i++) {

```

```
147         for(j = 0; j < n; j++) {
148             cin >> a[i][j];
149         }
150     }
151
152     // Processing
153     s = 0;
154     for (i = 0; i < m; i++) {
155         s += minDong(a[i], n);
156     }
157
158     // Print output
159     cout << round(1.0 * s / m);
160
161     return 0;
162 }
163
164
165
```