### 第一章

##### 习题1 - 5：打折（discount）

一件衣服95元，若消除满300元，可打八五折。输入购买衣服件数，输出需要支付的金额（单位：元），保留两位小数。

代码部分：

#include<iostream>

using namespace std;

int n, price = 95;

double sale, discount = 0.85;

int main()

{

printf("请输入购买衣服的数量n (输入0停止)\n");

while (scanf\_s("%d", &n) && n)

{

if (n \* price > 300) sale = n \* price \* discount;

else sale = n \* price;

printf("支付金额:%.2lf元\n\n", sale);

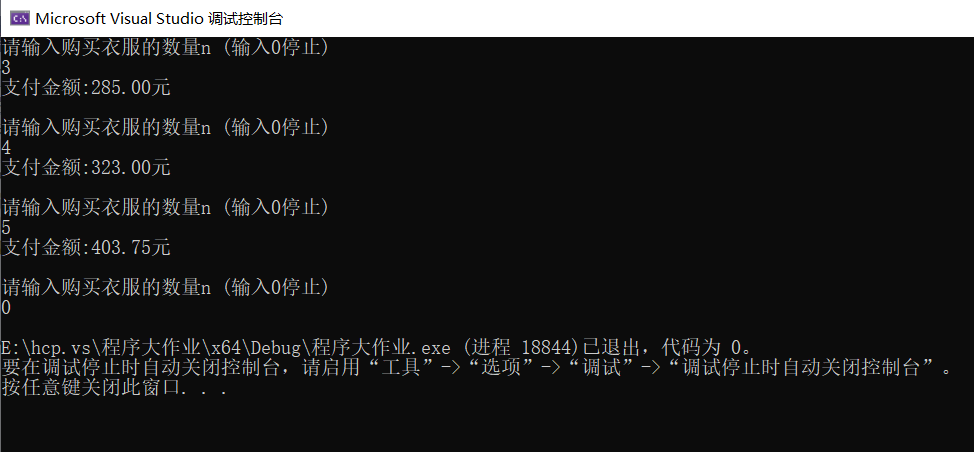
printf("请输入购买衣服的数量n (输入0停止)\n");

}

return 0;

}

运行结果：



### 第二章

##### 习题2 - 3：倒三角形（triangle）

输入正整数n <= 20，输出一个n层的倒三角形。例如，n = 5 时输出如下：

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

代码部分：

#include<iostream>

using namespace std;

int n;

void prin(int n)

{

int m = 1 + (n - 1) \* 2;

for (int i = 0; i < n; i++)

{

for (int k = 0; k < i; k++) cout << " ";

for (int j = 0; j < m - (i \* 2); j++) cout << "\*";

cout << endl;

}

}

int main()

{

printf("输入n：(n=0停止)\n");

while (cin >> n && n)

{

prin(n);

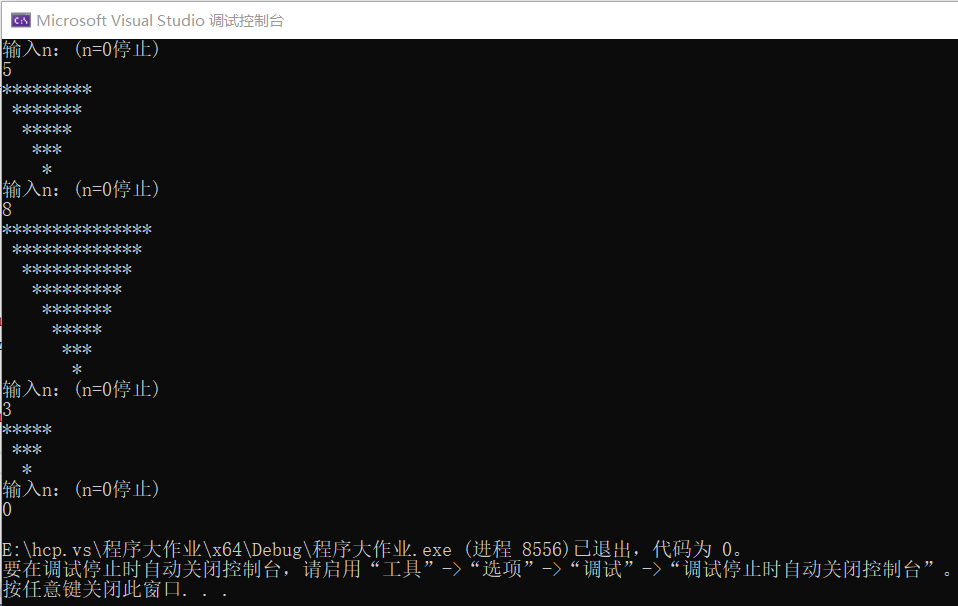
printf("输入n：(n=0停止)\n");

}

return 0;

}

运行结果：



### 第三章

##### 习题3 - 3：数数字（Digit Counting）

把前 n (n <= 1000) 个整数顺次写在一起；123456789101112...数一数 0 ~ 9 各出现多少次（输出10个整数，分别是0，1, ...，9出现的次数）。

代码部分：

#include<iostream>

using namespace std;

int n;

int main()

{

cout << "输入n(n <= 1000):";

while (cin >> n && n <= 1000)

{

string str;

int cnt[10] = { 0 };

for (int i = 1; i <= n; i++)

{

int a = i;

while (a)

{

char b = a % 10 + '0';

a /= 10;

str += b;

}

}

for (int i = 0; i < str.size(); i++)

{

cnt[str[i] - '0'] ++;

}

for (int i = 0; i < 10; i++) cout << cnt[i] << " ";

cout << endl << endl;

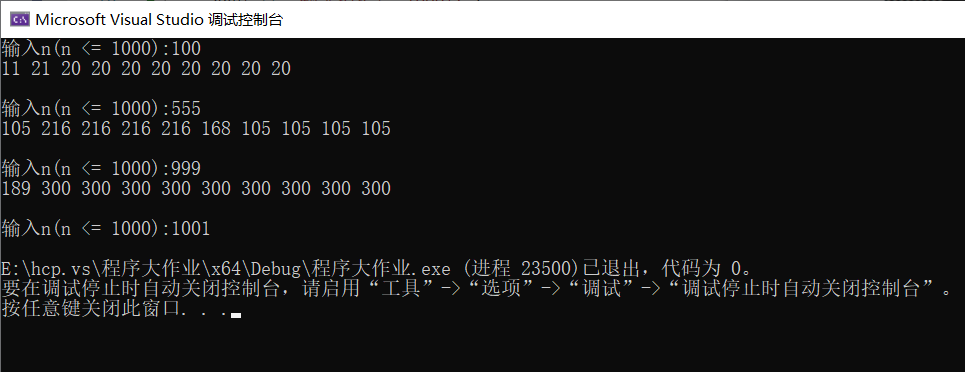
cout << "输入n(n <= 1000):";

}

return 0;

}

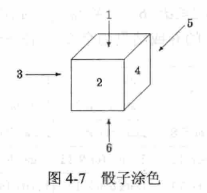
运行结果：



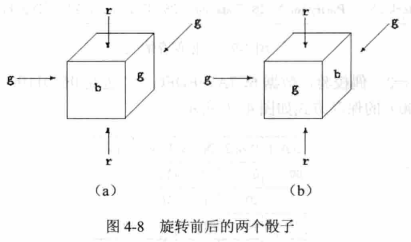
### 第四章

##### 习题4 - 4：骰子涂色（Cube painting）

输入两个骰子，判断二者是否等价。每个骰子用6个字母表示，如图4 - 7所示。



例如rbgggr和rggbgr分别表示图4 - 8所示的两个骰子。二者是等价的，因为图4 - 8（a）所示的骰子沿着竖直轴旋转90°之后就可以得到图4 - 8（b）所示的骰子。



代码部分：

#include<iostream>

using namespace std;

const int N = 10;

char s[2 \* N], a[N], b[N];

int main()

{

while (scanf\_s("%s", s, 20) != EOF)

{

for (int i = 0; i < 6; i++)

{

a[i] = s[i];

b[i] = s[i + 6];

}

int cnt = 0;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 6; j++)

{

if (a[i] == b[j] && a[5 - i] == b[5 - j])

{

cnt++;

a[i] = a[5 - i] = '0';

b[j] = b[5 - j] = '1';

break;

}

}

}

if (cnt == 3) cout << "TRUE" << endl;

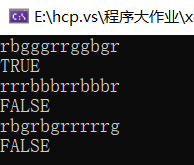
else cout << "FALSE" << endl;

}

return 0;

}

运行结果：



### 第五章

##### 习题5 - 4：交换学生（Foreign Exchange）

有n (1 <= n <= 500000)个学生想交换到其他学校学习。为了简单起见，规定每个想从A学校交换到B学校的学生必须找一个想从B换到A的“搭档”。如果每个人都能找到搭档（一个人不能当多个人的搭档），学校就会同意他们交换。每个学生用两个整数A、B表示，你的任务是判断交换是否可以进行。

代码部分：

#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

int main()

{

int n;

cout << "输入n(n!=0):";

while (cin >> n && n)

{

vector<int> sa, sb;

for (int i = 0; i < n; i++)

{

int a, b;

cin >> a >> b;

sa.push\_back(a);

sb.push\_back(b);

}

sort(sa.begin(), sa.end());

sort(sb.begin(), sb.end());

int flag = 1;

for (int i = 0; i < n; i++)

{

if (sa[i] != sb[i])

{

flag = 0; break;

}

}

if (flag) cout << "是否可以交换:Yes" << endl << endl;

else cout << "是否可以交换:No" << endl << endl;

sa.clear();

sb.clear();

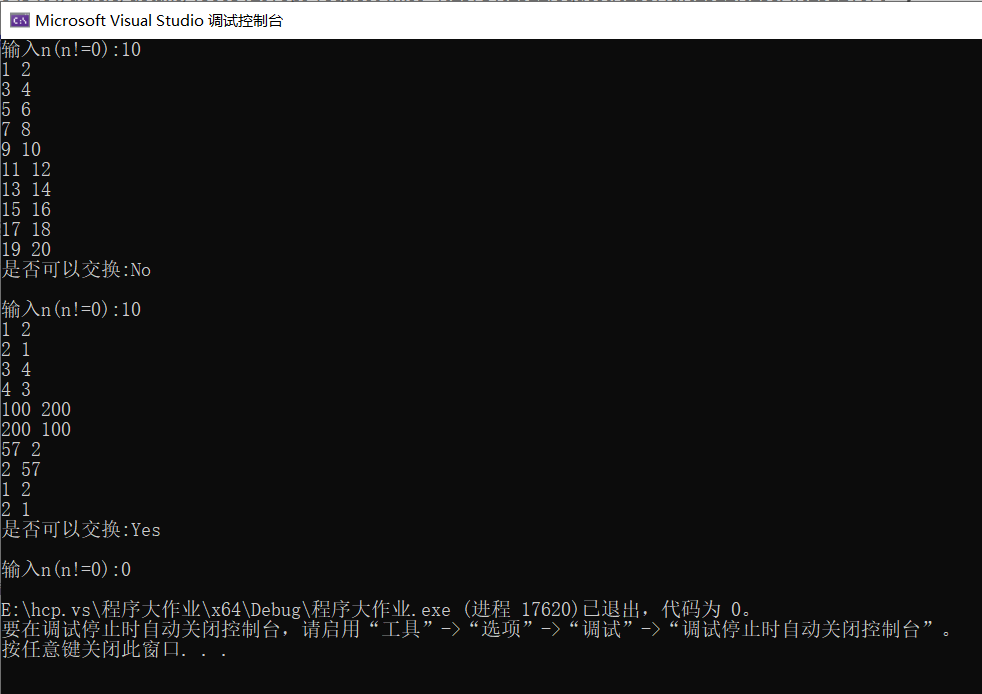
cout << "输入n(n!=0):";

}

return 0;

}

运行结果：



### 第六章

##### 习题6 - 1：平衡的括号（Parentheses Balance）

输入一个包含“（）”和“[]”的括号序列，判断是否合法。具体规则如下：

* 空串合法。
* 如果A和B都合法，则AB合法。
* 如果A合法则 (A) 和 [A] 都合法。

代码部分：

#include<iostream>

#include<stack>

using namespace std;

int n;

int main()

{

cin >> n;

cin.get();

while (n--)

{

stack<char> str;

char a;

int flag = 1;

while (cin.get(a) && a != '\n')

{

if (a == ')' && flag)

{

if (str.empty()) flag = 0;

else if (str.top() == '(') str.pop();

else flag = 0;

}

else if (a == ']' && flag)

{

if (str.empty()) flag = 0;

else if (str.top() == '[') str.pop();

else flag = 0;

}

else str.push(a);

}

if (str.empty() && flag) cout << "Yes" << endl;

else cout << "No" << endl;

}

return 0;

}

运行结果：

