# AA

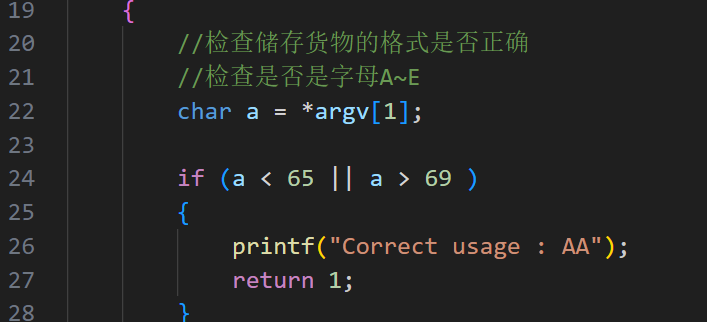
前言：

LEVEL 1-1

考虑到该level级别只用储存一种货物在一种通道，且无需重复购买，那么最好让用户第一次输入是储存货物而不是购买货物，于是我使用了argc与argv而不是int main(void),并写下部分代码来对用户的第一次输入进行限定。既然程序的第一步应该是对用户的输入进行检定，并判断其是否符合储存货物的格式，于是有了下面的代码：

<Level1_1_1.c>（不完整）

1. 如下图。首先，考虑到储存货物的格式为 ./xxx A 1 2 3，那么要如何判断argv[1]为大写字母呢？ 我联想到 在ASC II表中A~Z都对应一个数字，于是我选择用数字大小来作判断，但实际上argv[?]是一个指针，因此我要先取出其指向的字母，所以有了下图第22行的代码



图L1-1 (1)

1. 如下图。 接着，我需要检定argv[2]、argv[3]、argv[4]是否为整数。

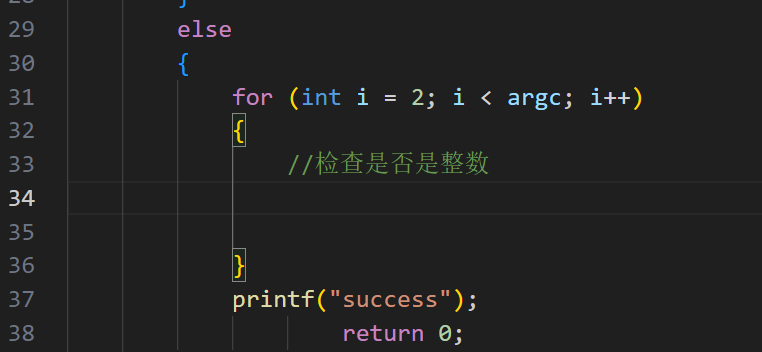


图 L 1-1 (2)

难点 1-1-1但是在如何检定其是否为整数上我有了疑问，通过上网搜索，发现ctype.h中的isdigit函数可以完成该任务，于是代码变成下图所示

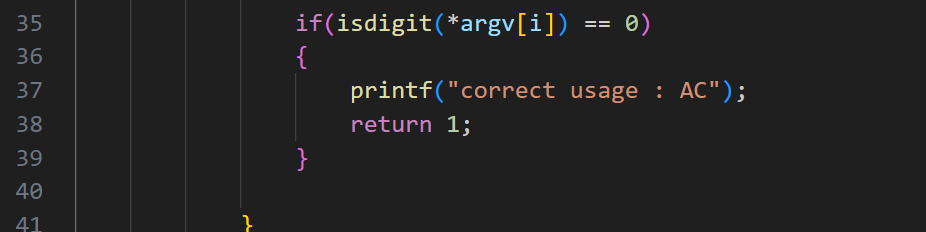


图 L 1-1 (3)

1. 之后还要依次判断输入的argv大小是否在给定区间以及输出货物情况，这一块我没有什么疑惑，完成后的部分如下部分链接所示：

<Level1_1_2.c>

LEVEL 1-2

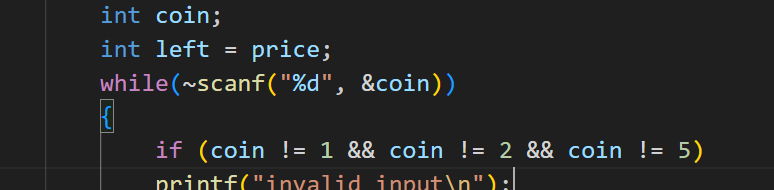
（投币部分）

难点1-2-1 在这一部分，对我个人而言最难的是如何反复监测用户的输入。

从最开始的储存货物，到输入总价以确定购买的商品种类，再到反复投币，让如何记录下用户的多次输入成为了一个难题。

事实上，在做这个任务之前，我只处理过只需要用户输入一次的任务。为了解决这个难题，我询问了高年级的学长，并学习到将while和scanf结合起来便可，

如下图所示



完整版LEVEL 1-2代码如下链接所示：<Level1_1_3.c>