Voj网址：<https://cn.vjudge.net/group/challenge> 英文题目

洛谷，牛客网 中文题目

通过汇编语言了解程序的实际构成：

本地代码的指令中，表示其功能的英语缩写称为助记符。汇编语言是通过助记符来记述程序的。

汇编语言的源代码转换成本地代码的方式称为汇编。汇编是使用汇编器这个工具来实现汇编的。

本地代码转换成汇编语言的源代码称为反汇编。通过反汇编，可以得到人们可以理解的代码。汇编语言的源文件的拓展名通常是.asm格式 .asm是assembler（汇编器）的缩写

汇编语言中的段定义是构成程序的命令和数据的集合组。在高级编程语言的源代码中，即使指令和数据在编写时时分散的，编译后也会在段定义中集合汇总起来。

汇编语言的跳转指令是在将程序流程跳转到其他地址时需要用到的指令。在汇编语言中，通过跳转指令，可以实现循环和条件分支

汇编语言和本地代码是一一对应的：

比如add(addition)cmp(compare)称为助记符，使用助记符的编程语言称为汇编语言。

无论什么语言，反汇编后得到的代码都是汇编语言的代码。但是，比如使用c语言变成的本地代码进行反汇编，再得到原始的c语言的代码是不太可能的，因为c语言和本地代码不是一一对应的。

在c语言中，分号表示一行代码的结束；在汇编语言中，分号（；）表示的是注释行，在汇编语言中，一句话占用一行。

不会转换成本地代码的伪指令：

汇编语言的源代码，是由转换成本地代码的指令（操作码）和针对汇编器的伪指令构成的。伪指令负责把程序的构造及汇编的方法指示给汇编器（转换程序），不过伪指令本身是无法汇编转换成本地代码的。

段定义：段定义是用来区分或者划定范围区域的意思。在汇编语言中segment伪指令表示段定义的起始，ends伪指令表示段定义的结束。段定义是一个连续的内存空间。

伪指令proc和endp围起来的部分，表示的是过程的范围。

汇编语言的语法是：操作码加操作数

在汇编语言中，一行表示对汇编语言的一个指令。类似于mov这样的指令称为“操作码”（opcode），作为指令对象的内存地址以及寄存器称为“操作数”（operand）。操作码表示的是指令动作，操作数表示的是指令的对象。

操作码和操作数罗列在一起的语法，就像是一个英文的指令文本，操作码是动词，操作数是宾语。

Push和pop是汇编语言中入栈和出栈的指令。

函数的参数是通过栈来传递的，返回值是通过寄存器来返回的。

再提一次，在c语言中，在函数外部定义的叫做全局变量，在函数内部定义的叫做局部变量。

在汇编语言中，标签表示的是相对于段定义起始位置的位置。

局部变量只能在定义该变量的函数内使用是因为局部变量是临时表保存在寄存器和栈中的。

函数内部利用的栈，在函数处理完毕后会恢复到初始状态，因此局部变量的值也被销毁了，寄存器也可能被用于其他的目的。总的来说，局部变量只是在函数运行处理的时候临时存储在寄存器和栈上。保存局部变量的是寄存器和栈，一般寄存器空闲的情况下使用寄存器，寄存器不空闲的情况下使用栈。

在汇编语言中，跳转指令有：jle（jump on less or equal）（如果小于或者等于就跳转）；

Jmp（无条件跳转）；jge（jump on greater or equal）（如果大于或者等于就跳转）

多线程处理：线程是操作系统分配给CPU的最小运行单位，源代码的一个函数就相当于一个线程。多线程处理指的是一个程序中同时运行多个函数的意思。

硬件控制方法：

在汇编语言中，是通过IN指令和OUT指令和外围设备进行输入输出操作的。比如在x86系列的CPU中，通过IN指令实现I/O输入。通过OUT指令实现I/O的输出。I/O是电脑主机和外围设备进行交互的。

IN指令就是通过指定的端口号的端口输入数据，并且将它存储在CPU内部的寄存器中。OUT指令则是把CPU寄存器中存储的数据，输出到指定的端口号的端口。

用来识别外围设备的编号的称为I/O地址或I/O端口号

DMA（direct memory access）：不经过CPU中介的处理，外围设别直接同计算机的主内存进行数据传输。用来识别具有DMA功能的外围设备是DMA通道，比如磁盘这样用来处理大量数据的外围设备都具有DMA功能。

在c语言等高级编程语言开发的Windows应用中，大家很少能够接触到直接控制硬件的指令，这是因为硬件的控制是由windows全权负责的。

系统调用：windows提供了通过应用来间接控制硬件的办法。利用操作系统提供的系统调用功能就可以实现对硬件的控制。在Windows中，系统调用称为API。各API就是应用调用的函数，这些函数的实体被存储在DLL文件中。

端口号和端口：

在计算机主机中，附带了用来连接显示器及键盘等外围设备的连接器，各连接器的内部都连接了用来交换计算机主机同外围设备之间电流特性的IC，这些IC统称为I/O控制器。由于电压不同，数字信号及模拟信号的电流特性也不同，计算机主机和外围设备是无法直接连接的。为了解决这一个问题，I/O控制器就变得很有必要了。

端口：显示器，键盘等外围设备都有各自专门的的I/O控制器。I/O控制器中就有用于临时保存输入输出数据的内存，这个内存就是端口。

端口（port）的字面意思是港口，因为端口就像是计算机主机与外围设备之间的进行货物（数据）装卸的港口。I/O空机器内部的内存，也称为寄存器。当然了I/O内存中的寄存器和CPU中的寄存器是不一样的，CPU中的寄存器是用来进行数据运算处理的，而I/O寄存器则主要是用来临时存储数据的。

在实现I/O控制器功能的IC中，会有很多个端口。由于计算机中连接着很多的设备，所以就会有很多个I/O控制器，当然也就会存在很多个端口。

一个I/O控制器既可以控制一个外围设备，也可以控制多个外围设备。各个端口之间通过端口号进行区分。端口号也称为I/O地址，IN指令和OUT指令在端口和CPU之间进行数据和输入输出。这和通过内存地址来进行主内存的读写是一样的道理。

I/O装置，有的直接附带在计算机主机的主板上（用来放置CPU的基板），有的则是各自独立的拓展板卡。键盘，鼠标，打印机等常用的I/O，一般都在主板上，而输出显示图形的先啦以及网卡等特殊的I/O，通常是独立的拓展板卡。

键盘通过用于键盘的I/O（端口号为XXXX或者YYYY等之类的），转换成电流特性的转换，通过端口存入CPU的寄存器，寄存器再通过端口把信息转换成电流特性的转换输出到显示器等外围设备上面。

在Windows系统中通过控制面板—系统安全—系统—设备管理查看软盘驱动控制器的属性时的情况，“I/O的范围”右侧的数值就是端口号。通过指定该端口号，并利用IN/OUT命令，就可以直接控制软驱这个硬件设备，实现输入输出处理了。

外围设备：把这个东西取出，对计算机主机没有什么影响，包括蜂鸣器，它也是外围设备的一种。

在c语言中插入汇编语言助记符：

在大部分C语言的处理（编译器的种类）中，只要使用\_asm{}括起来，就可以在其中记述助记符。也就是说，这样就可以编写C语言和汇编语言混合的源代码。

助记符不区分大小写指令，有些c语言和汇编语言的结合体在Windowsxp开始（当然win10是非常先进的系统了）就不可以正常运行了。原因是现在的windows禁止了应用直接控制硬件的方式。

外围设备的中断请求：

IRQ（interrupt request）是终端请求的意思。一般在”I/O范围“下面有一个”IRQ“项目。IRQ是用来中断当前正在运行的程序，并且跳转到其他程序运行的必要机制。这个机制称为中断处理。中断处理在硬件控制中担当着重要角色。因为如果没有中断处理，就有可能出现处理无法顺畅运行的情况。

中断信号：

实施中断请求的是连接外围设备的I/O控制器，负责实施中断处理程序的是CPU。为了进行区分，外围设备的中断请求会使用不同于I/O端口的其他编号，该标号称为中断信号。

顺带再提：BIOS位于计算机主板或者拓展板卡上内置的ROM中，里面记录了用来控制外围设备的程序和数据。

如果同时有多个外围设备进行中断请求的话，CPU会为难。因此，再I/O控制器和CPU之间加入中断控制器的IC来进行缓冲。中断控制器会把从多个外围设备发出的中断请求有序的传递给CPU。

中断请求的顺序一般是这样的：

主程序的处理--发生中断请求--中断处理程序—寄存器的备份—外围设备的控制—寄存器的还原—主程序的处理

轮询：由于外部设备有很多个，因此就有必要按照顺序来调查。按照顺序调查多个外围设备的状态称为轮询。

DMA可以实现短时间内传送大量数据：

再次重申：DMA指的是在不通过CPU的情况下外围设备直接和主内存进行数据传送。磁盘等都用到了这一个机制。通过利用DMA，大量的数据可以很快速的传送给主内存。DMA快速的原因就是CPU作为中介的时间被节省了。

在资源标签中有DMA设定，该设定编号称为DMA通道。CPU借助DMA通道，来识别是哪一个外围设备使用了DMA。

注：端口号，IRQ，DMA等可以指定的数值范围是有限的，因此它们也是资源的一种。

文字及图片的显示机制：

显示器中显示的信息一直存储在某内存中。该内存称为VRAM（video RAM）。在程序中，只要往VRAM中写入数据，该数据就会在显示器中显示出来。实现该功能的程序，是由操作系统或BIOS提供，并借助中断来处理的。

在现在的计算机中，显卡等专用硬件中一般都配置有与主内存相独立的VRAM和GPU（图像处理器也成为图形芯片graphics processing unit）。因为对于经常需要描述图形的Windows来说，一个很大的VRAM是必要的。为了提升图形的描绘速度，有时候还需要专用的图形处理器。但不管怎么样，内存VRAM中存储的数据就是显示器上显示的信息，这个机制是不变的。

事实上，无论程序的内容是什么，计算机做的事情永远只是对输入的数据内容进行运算，然后把运算好的结果输出，这一点是不会发生改变的。

‘

让计算机思考：

用计算机进行的模拟实验叫做：计算机模拟。计算机模拟指的是用软件来进行实际实验。

伪随机数指的是通过公式产生的伪随机数。伪随机数同真正的随机数不同，伪随机数存在周期。

随机数的种子是生成伪随机数的公式中使用的参数，随机数的种子不同，产生随机数也是不同的。

至于计算机，暂时来讲是没有思考功能的。所谓的人工智能，就是通过随机数的方式让电脑给人一种电脑有思想的感觉，即人工智能。比如和电脑的剪刀石头布的游戏。

AI（artificial intelligence）是人工智能的意思。

作为“工具“的程序和为了”思考“的程序：

程序就像是计算机执行各种指令罗列起来的文章，计算机内部的CPU，通过对该文章的内容进行解析和运行，来控制连接到计算机的各种外围设备。控制就是指CPU在各种设备直接按配合进行数据的输入输出的处理。

程序的使用目的大致分为两类：一类是大家作为工具来使用的程序，比如文字处理器等；另外一种是用程序来代替执行人类的思考过程。比如电饭煲里面的微计算机根据米和水的分量来调整火的大小以及加热的时间，以做出好吃的饭。这种事情由计算机代替执行了，借助这种程序，使计算机有了“思考“的能力。

统一随机数：就是我们常常讲的随机数。在c语言中，rand（）函数的返回值就是统一随机数。

使用c语言中的随机数：

首先使用头文件#include<stdlib.h>

然后 srand(time(NULL));//设定随机数的种子

Int a=rand()%10;//获取随机数

类似于以上这样就可以把一个0到10的随机量赋值给变量a

程序生成随机数的办法：

计算机模拟指的是利用计算机模拟实际实验的方式。经常用于建筑物的耐震实验等实际难以进行的实验中。使用随机数的计算机模拟有时候也称为“蒙特卡洛法“，来源于因为赌博而著名的城市”蒙特卡洛“。

随机数使表示人类的直觉以及念头的一种办法，

随机色子数是用来产生随机数的一种工具，每一个色子有20个面晃动随机数色子后，出现在正面的数字就是随机数。由于计算机不能晃动随机数色子，因此程序一般会通过生成类似于随机数的数值公式来得到随机数。在c语言中，虽然公式的实体是被隐藏的，但是只要调用rand（）函数，就可以得到结果（随机数）。不过由于借助公式产生的随机数具有一定的规律性，因此并不是真正的随机数，它们通常称为伪随机数。虽然是伪随机数，但还是非常有用的。

一个获取伪随机数的方式：线性同余法

在获取随机数的时候公式里面的数值（事先存在的）称为随机数的种子

灵活使用记忆功能以达到更加接近人类的标准：

直觉并不是简简单单的简单思考，通常还带有一定的思维习惯。

两个索引组成的数组叫做二维数组。

用程序来表示思维模式：

在很多人的大脑中，在进行剪刀石头布的时候，往往不是每种可能都是1/3的。有的人可能更加偏向于石头，有的人可能大脑中一直存在剪刀石头布的主旋律……根据人类的这些思维方式，制定合适的策略，会大大增加多次实验中计算机获胜的几率。