同济大学

汇编语言程序设计

（上机大作业报告）

学　号　：**1853971**

姓　名　：王天

专业/年级：计算机科学与技术/大二

所在院系　：电信学院计算机系

任课教师　：朱珏

完成日期　：**2020 年 8 月 3 日**

一.设计题目

某学生利用暑期到快递公司打工，该公司以底薪加计件工资的形式，并以周为结算周期给实习学生发工资。具体计薪办法是：实习学生一周工作 6 天，每天基本工资 220 元，每天送快递 100 件为基本要求，超过 100 件的则每件增加 1.5 元，每天不足 100 件则每少 1 件扣 1.2 元。根据某同学某周内各天的快递量（如 102,176,147,89,138,72）编程计算该实习学生的周工资。

二.设计说明

1.整个程序从键盘输入各天的快递量并判断其合法性，每天基本工资 220 元，以送快递 100 件为基本要求，超过 100 件的则每件增加 1.5 元，每天不足 100 件则每少 1 件扣 1.2 元，计算各天的日工资并将其累加得到周工资，最终将周工资保留一位小数在屏幕上显示出来。输入和输出时均在屏幕上给出相应的提示信息和错误信息。

2.整个工作包括 MAIN、INPUT、SOLVE、OUTPUT 和 DISP 共 5 个任务(子程序)，MAIN 调用 INPUT 完成输入相关工作，调用 SOLVE 完成日工资和周工资的计算，调用 OUTPUT 完成输出相关工作，INPUT、OUTPUT 均调用 DISP 用于在屏幕上显示十进制数字。

3.各子程序功能及参数传递如下：

（1）MAIN：主程序，通过调用 INPUT、SOLVE、OUTPUT 完成程序的整体功能。无入口参数和出口参数。

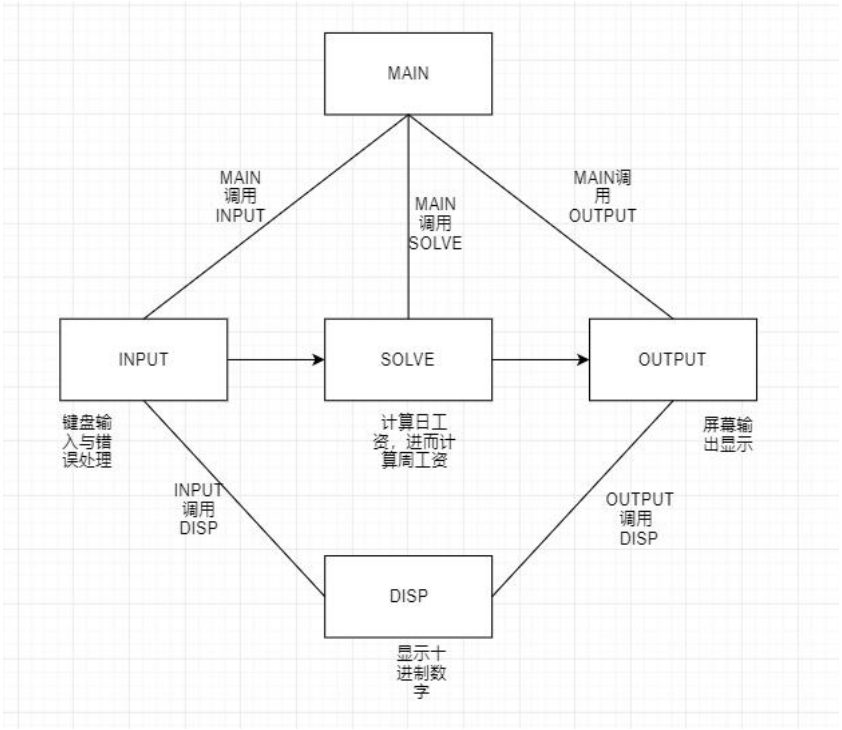
（2）INPUT：给出输入提示信息，完成数据从键盘的输入，判断输入数据的合理性并对错误数据给出错误信息。无入口参数,出口参数为 NUM(存储每天快递量)，即将所输入数据存储于 NUM 数组中。

（3）SOLVE：对 NUM 数组中的各天快递量，按照每天基本工资 220 元，以送快递 100 件为基本要求，超过 100 件的则每件增加 1.5 元，每天不足 100 件则每少 1 件扣 1.2 元的规则计算每日工资，并将每日工资累加得到周工资。入口参数为 BX(NUM 数组首地址)，出口参数为 ANS(周工资)。

（4）OUTPUT：给出输出提示信息，将周工资以保留 1 位小数的各式输出在屏幕上。入口参数为 ANS(周工资)，无出口参数。

（5）DISP：将寄存器中的二进制数据以十进制的形式在屏幕上显示出来。入口参数为 AX(要显示的数据)，无出口参数。

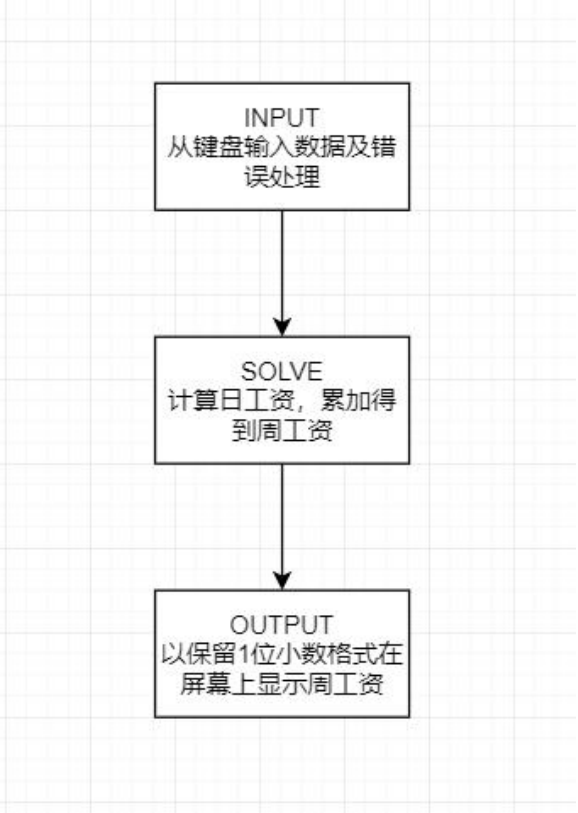
4.程序框图如下：



5.各子程序详细设计说明如下：

（1）MAIN：主程序。首先调用 INPUT 子程序完成从键盘的数据输入和错误处理，得到各天的快递量，存于 NUM 数组中。其次调用 SOLVE 子程序根据 NUM 数组的数据计算出对应的日工资，并在此过程中将日工资累加得到周工资，存于 ANS 中。最后，调用 OUTPUT 子程序将 ANS 中对应的数据转化成保留 1 位小数的格式输出在屏幕上。

流程图如下所示：



（2）INPUT：输入部分子程序。首先将 NUM 数组的首地址送入 BX 寄存器，使得 BX 起到指示当前数组元素位置的作用，之后利用条件循环，在 CX 寄存器数据不超过总天数 N 的情况下进行各天的快递量输入，超过总天数 N 则子程序调用结束并返回。

在从键盘输入数据之前，先通过 DOS 的 9 号功能调用进行字符串输出，将事先存储的提示信息显示在屏幕上(其中涉及到的当前天数信息需要根据 CX 寄存器中的值调用 DISP 显示)。

对于每一天的输入，通过 DOS 的 1 号功能调用读取键盘输入的每个字符。

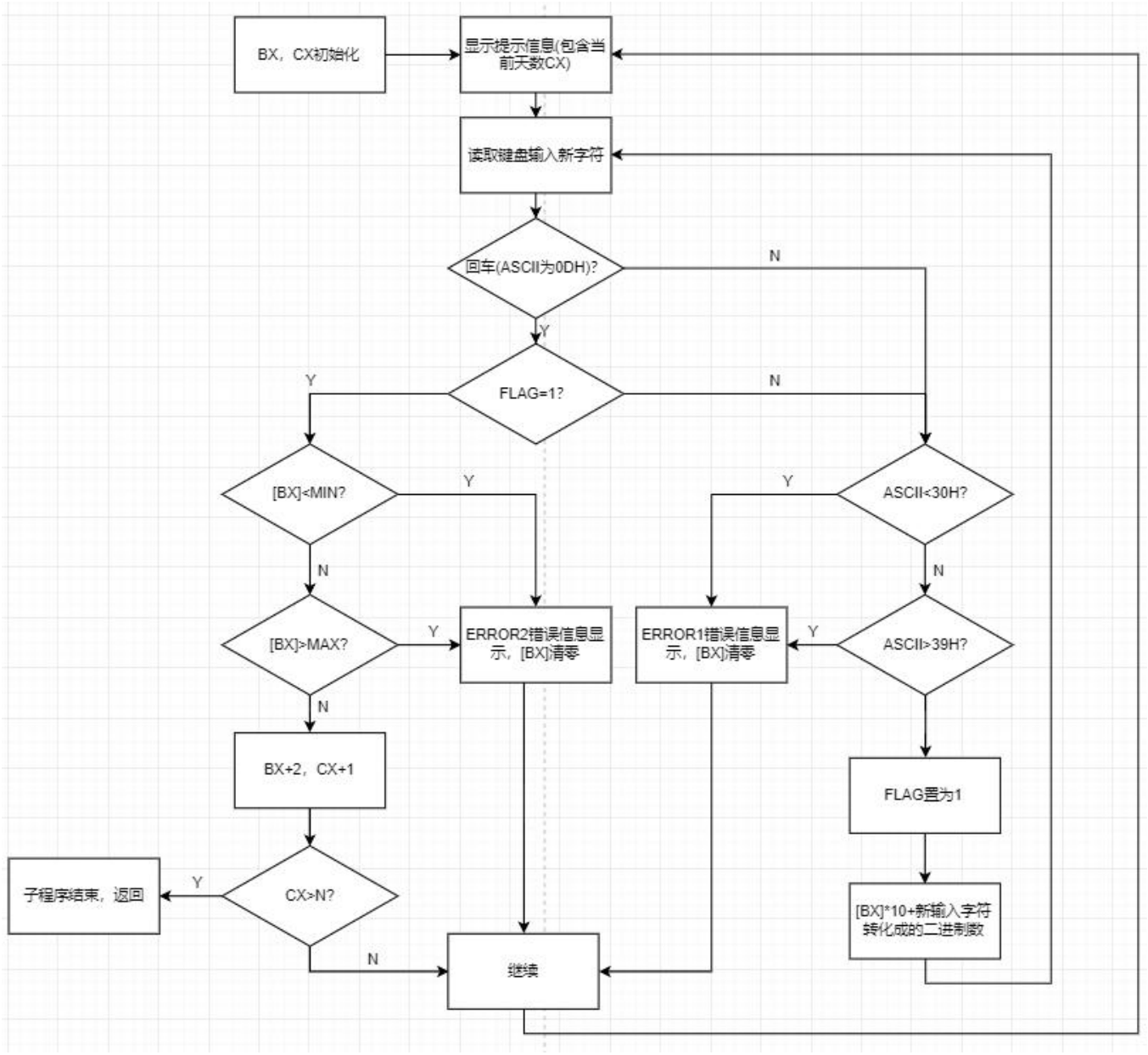
读取到回车字符时进行判断：若 FLAG 变量的值为 1，则此前读取过数字字符，结束读取；若 FLAG 变量的值为 0，则此前未读取过数字字符，继续读取。

读取到非数字字符时(ASCII 码值大于 39H 或小于 30H)，将错误信息显示在屏幕上并清零当前数组元素，之后重新进行当天的快递量输入。

读取到数字字符时，将输入字符转化成二进制数，并利用当前数组元素内容*10+新二进制数的方式将新输入数字拼接到已输入数字中，从而更新当前数组元素保存的当前快递量。

当一天的快递量输入结束后，将数组元素中数据视为无符号数，判断快递量是否在[MIN,MAX]的合理区间范围内：若不在范围内则将错误信息显示在屏幕上并清零当前数组元素，之后重新进行当天的快递量输入；若在范围内则将 BX、CX 的值进行更新，便于继续输入之后的快递量。

流程图如下所示：



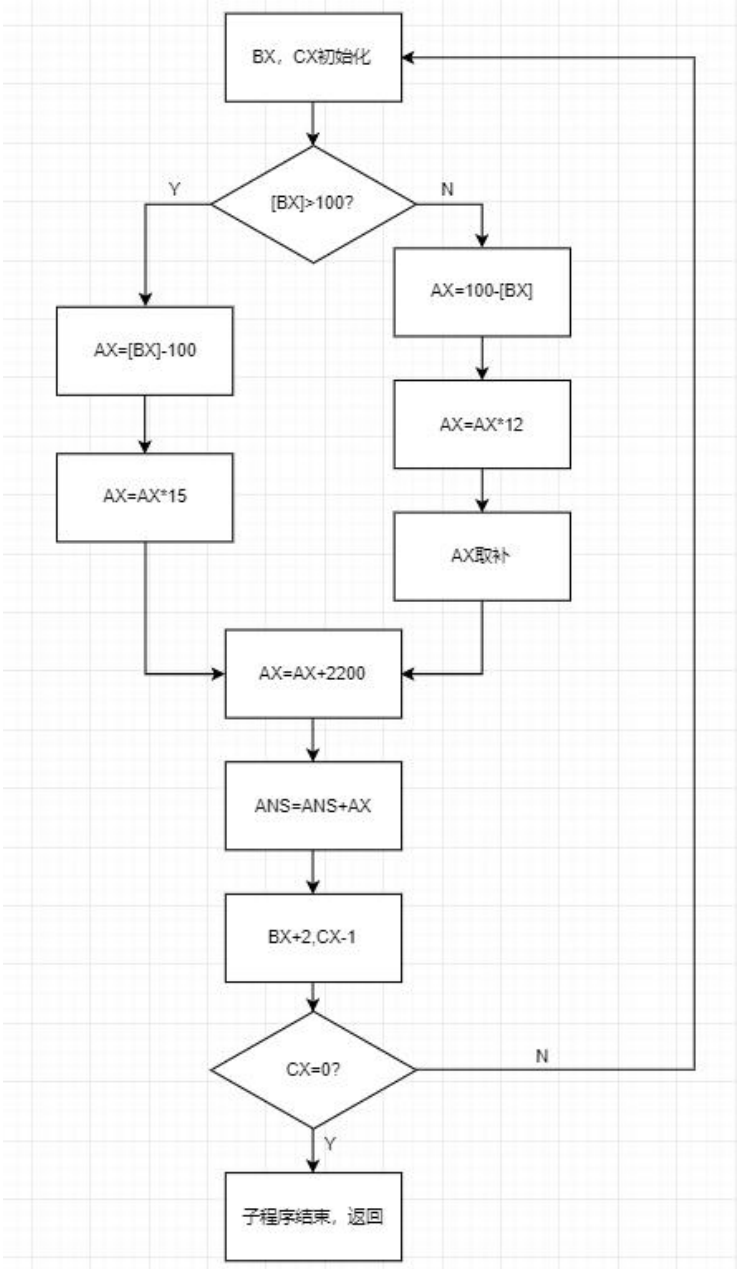
（3）SOLVE：计算部分子程序。首先将 NUM 数组的首地址送入 BX 寄存器，使得 BX 起到指示当前数组元素位置的作用，将 CX 寄存器预置为 N，之后利用计数循环 LOOP，进行各天的工资的计算并累加，当 CX 变为 0 时则子程序调用结束并返回。

对于每一个当前元素值，判断其大于 100 或小于等于 100：若其大于 100，则说明需要对超出部分进行奖励，将元素值减去 100 存于 AX 中，之后将 AX 的值乘以 1.5；若其小于 100，则将 100 减去元素值存于 AX 中，之后将 AX 的值乘以 1.2，再将 AX 取补(相当于取相反数，将减法运算统一为加法运算)。还需要再加上基本工资，因此还要将 AX 加上 220。

计算完当前元素对应的日工资后，将 ANS 加上 AX 寄存器中的值以累加计算周工资，并将 BX、CX 的值进行更新，便于继续计算之后的日工资。

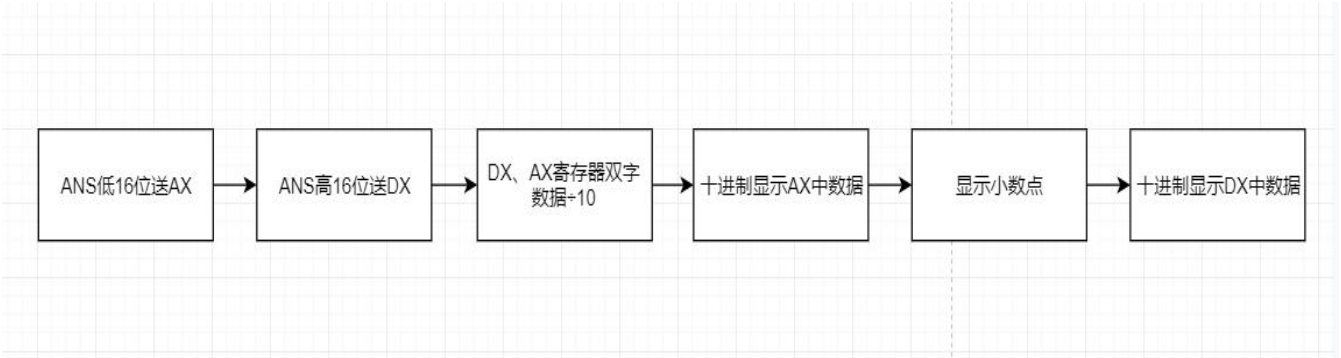
实际操作时为避免浮点数运算，采取将数据扩大 10 倍的方法，实际是分别乘以 15 和 12、加上 2200。同时由于扩大了 10 倍进行计算，虽然日工资不会超过字类型有符号整数范围，但累加结果可能会超过，因此 ANS 需要使用双字类型进行存储(进行加法运算时也要高低位分别运算)。

流程图如下所示：



（4）OUTPUT：输出部分子程序。首先通过 DOS 的 9 号功能调用在屏幕上显示输出提示信息。之后将双字变量 ANS 的高 16 位送入 DX 寄存器，低 16 位送入 AX 寄存器。由于在 SOLVE 子程序中计算工资时均扩大了 10 倍，因此将 DX、AX 共同组成的双字数据除以 10，在 AX 中得到商，DX 中得到余数。此时 AX 中即为工资的整数部分，DX 中即为工资的 1 位小数。调用第一次次 DISP 子程序，将 AX 中数据以十进制显示在屏幕上，再通过 DOS 的 2 号功能调用在屏幕上显示小数点，最后第二次调用 DISP 子程序，将 DX 中数据以十进制显示在屏幕上。

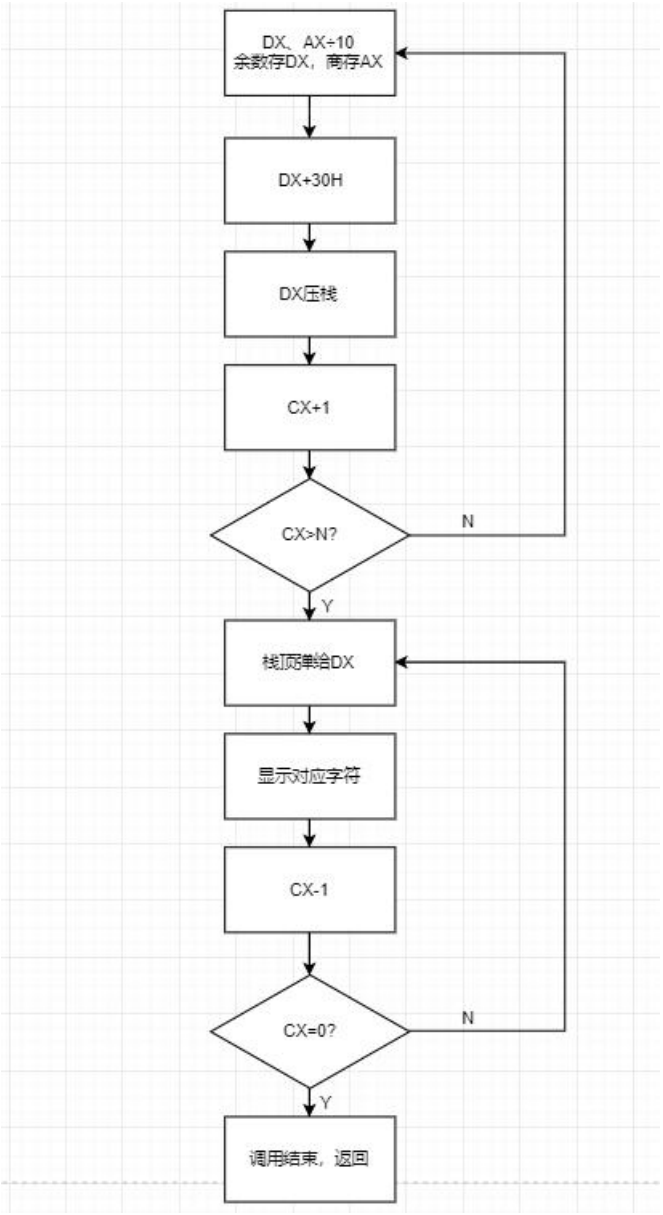
流程图如下所示：



（5）DISP：十进制显示子程序。利用 AX 寄存器传递参数，存储要以十进制显示在屏幕上的数据。之后利用条件循环，每次循环时将 DX 和 AX 中数据除以 10，商留在 AX 中，余数保存在 DX 中。将 DX 中数据加上 30H 转化成对应数字字符的 ASCII 码，并压栈保存，从而使得 AX 中数据十进制形式从低位到高位依次入栈。每压一次栈 CX 就记一次数。每次循环结束时判断：若 AX 不为 0 则继续循环，否则结束循环。

根据 CX 记录的压栈次数，使用 LOOP 进行计数循环，每次循环从栈顶弹出一个字到 DX 中，通过 DOS 的 2 号功能调用显示出对应的数字字符。循环结束时，原 AX 寄存器对应的数据即按照从高位到低位的顺序显示在屏幕上。

流程图如下所示：



三. 调试说明

1. 调试情况

（1）题目中限定了计算 6 天的总工资这一条件，并且工资的计算方法中基准工资、超出部分每件奖励、不足部分每件惩罚都是给定的数据，额外补充的快递量范围也已经限定。这些数据并不是永远不变的，将这些数据根据实际情况进行调整，可以得到和题目相似的一系列问题。为了解决具有共性的一类问题而不仅仅是解决一道题，对于这些可变数据不应该在代码段中直接写死，而应该在数据段中统一定义为一个名称，在需要修改这些数据时仅需要在数据段修改一处即可。例如下图所示：

```
N       EQU     6       ;统计的天数
STA     EQU     100     ;每日基准件数
BAS     EQU     2200    ;每日基本工资
MORE    EQU     15      ;超出部分每件奖励工资
LESS    EQU     12      ;不足部分每件扣除工资
MIN     EQU     0       ;件数下限
MAX     EQU     999     ;件数上限
```

当每周工作变为 5 天、基准件数变为 120 件、超出部分奖励工资变成 1 元等情况发生时，只需要将 N 更改为 5、STA 更改为 120、MORE 更改为 10 即可。如果是直接在代码段中写出 MOV CX,6 等语句，则每一处用到天数的地方都需要修改，较为繁琐。

使用 EQU 伪指令在不占用内存的情况下统一定义常量，即避免了在代码段写死数据带来修改的麻烦，也避免了定义变量存储数据(如 N DB 6 等)导致用到存储器操作数降低运算速度，是一种时空效率较高的解决方法。

（2）在设计 INPUT 子程序时，由于在提示信息中包含天数信息，需要按递增顺序显示出来，为了方便直接调用 DISP 子程序，就采用 CX 寄存器从 0 递增到 N 的条件循环控制方式，依次输入各天快递量。

这种情况下，刚开始我将 CX 初值置为 0，并在循环开始前判断 CX 是否大于 N，若大于则跳转到循环之外，进行后续处理。但是由于 JA 指令跳转偏移范围为-128~127 字节，而循环体较为复杂，其中包含的指令所占总字节数已经超过了 127 字节，使得在进行汇编时出现以下错误：

```
K1853971.ASM(63): error A2053: Jump out of range by 33 byte(s)

  49870 + 455343 Bytes symbol space free

       0 Warning Errors
       1 Severe  Errors
```

JA 指令无法直接跳转到过长的循环体之外，使得程序无法正常汇编。对于此问题，我尝试了两种解决方法。

第一种方法是使用间接跳转的方法，在跳转过程中采用多段跳转的方式，使得每一小段跳转范围都在-128~127 字节之间。这种方法会在程序中插入很多中间跳转语句，且每个中间跳转语句之前还需要一个保护跳转语句，用来避免程序顺序执行到中间跳转语句引起错误。从而导致程序的结构遭到一定程度的破坏，代码量也会增加。

第二种方法是将循环控制条件放在循环末尾处，即最开始将 CX 置为 1，每次执行完循环体后 CX+1，再判断 CX 是否大于 N。此时由于跳转语句也在循环体末尾，因此跳出循环之外所需的距离很短，JA 指令能够正常跳出循环。由于 N 至少为 1，因此循环体至少会执行一次，循环控制条件放在循环末尾不会影响运行结果。最终采用了这种解决方法。

（3）在程序设计时需要考虑到问题的可变性，不宜针对性过强而导致程序的适用范围很窄。且对于可选用的解决手段，需要比较优劣，采用最合适的方法。同时对于循环分支程序的设计，需要合理安排分支路线，条理清晰，避免跳转来跳转去的混乱场面。

2.连接的要求说明

本程序虽然采用子程序设计，但并没有采用模块化，所有子程序在同一个源文件中。连接时，只需要在 DOSBOX 中使用“link 1853971”命令即可完成连接。如下图所示。

```
C:\>link K1853971

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

Run File [K1853971.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
```

3.运行结果展示

（1）首先验证程序运行的正确性：

220*6+（2+76+47+38）*1.5-（11+28）*1.2=1517.7，程序运行结果正确，如下所示：

```
C:\>K1853971
Please input the NUM of day 1 (range from 0 to 999) : 102
Please input the NUM of day 2 (range from 0 to 999) : 176
Please input the NUM of day 3 (range from 0 to 999) : 147
Please input the NUM of day 4 (range from 0 to 999) : 89
Please input the NUM of day 5 (range from 0 to 999) : 138
Please input the NUM of day 6 (range from 0 to 999) : 72
The total wage is : 1517.7
C:\>_
```

（2）其次验证程序对边界数据的处理：

100 是本题计算的重要划分标准，220*6+（1+2）*1.5-（1+2）*1.2=1320.9，程序对于边界数据有较好处理能力，如下所示：

```
C:\>K1853971
Please input the NUM of day 1 (range from 0 to 999) : 100
Please input the NUM of day 2 (range from 0 to 999) : 100
Please input the NUM of day 3 (range from 0 to 999) : 99
Please input the NUM of day 4 (range from 0 to 999) : 101
Please input the NUM of day 5 (range from 0 to 999) : 102
Please input the NUM of day 6 (range from 0 to 999) : 98
The total wage is : 1320.9
C:\>_
```

（3）再验证程序在极端数据下是否会溢出：

（899*1.5+220）*6=9411，大数据下程序运行不会溢出,如下所示：

```
C:\>K1853971
Please input the NUM of day 1 (range from 0 to 999) : 999
Please input the NUM of day 2 (range from 0 to 999) : 999
Please input the NUM of day 3 (range from 0 to 999) : 999
Please input the NUM of day 4 (range from 0 to 999) : 999
Please input the NUM of day 5 (range from 0 to 999) : 999
Please input the NUM of day 6 (range from 0 to 999) : 999
The total wage is : 9411.0
C:\>
```

（220-100*1.2）*6=600，小数据下程序运行无异常，如下所示：

```
C:\>K1853971
Please input the NUM of day 1 (range from 0 to 999) : 0
Please input the NUM of day 2 (range from 0 to 999) : 0
Please input the NUM of day 3 (range from 0 to 999) : 0
Please input the NUM of day 4 (range from 0 to 999) : 0
Please input the NUM of day 5 (range from 0 to 999) : 0
Please input the NUM of day 6 (range from 0 to 999) : 0
The total wage is : 600.0
C:\>_
```

（4）最后验证程序对数据正确性判断：

每日快递量合理区间为[0，999]，547 在此范围内，继续输入第 2 天的快递量。8500 超出此范围，报错误信息，并重新输入第 2 天快递量。1000 也超出此范围，再次报错并重新输入。由于快递量非负，‘-’不是数字字符，报错误信息并重新输入。12 在此范围内，继续输入第 3 天快递量。仅输入回车时并不结束该天快递量输入，而是换行并继续等待快递量输入，直到输入 14 再回车才进入下一天的输入。程序对各类错误情况判断较为完善，如下所示：

```
C:\>K1853971
Please input the NUM of day 1 (range from 0 to 999) : 547
Please input the NUM of day 2 (range from 0 to 999) : 8500
Numbers exceeded out of range, please retry.
Please input the NUM of day 2 (range from 0 to 999) : 1000
Numbers exceeded out of range, please retry.
Please input the NUM of day 2 (range from 0 to 999) : -
Non-number characters contained, please retry.
Please input the NUM of day 2 (range from 0 to 999) : 12
Please input the NUM of day 3 (range from 0 to 999) :


14
Please input the NUM of day 4 (range from 0 to 999) :
```

4.运行结果分析

本程序对临界数据处理完善，能够判断数据正确性，在正确输入情况下均能得到正确计算结果，在错误输入情况下能够给出错误提示信息并要求重新输入。具体细节如前文所述，总体运行结果较好。

四.使用说明

1.程序在 Windows10 家庭中文版 64 位操作系统下，使用 Intel Core(TM) i5-7300HQ 处理器和 8G 内存，使用 DOSBox0.74 软件进行汇编、连接、运行操作，能够正确运行。其余环境不一定适用。

2.程序的使用方法

将 K1853971.asm 源文件置于 masm 文件夹中，运行 DOSBox0.74 软件，进行如下汇编、连接步骤：

```
C:\>masm K1853971
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

Object filename [K1853971.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  49870 + 455343 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link K1853971

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

Run File [K1853971.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

_
```

之后输入 K1853971.exe 即可运行程序，完成正常使用。

3.要求输入信息的类型与格式

（1）输入信息为十进制无符号非负整数，不能加前缀‘+’号，可以加前缀零

（2）要求输入的每天快递量在[0,999]范围内，否则会给出错误提示信息

（3）所输入信息仅包含‘0’~‘9’的数字字符和回车，否则会给出错误提示信息

4.出错信息的含义及注意事项

（1）"Non-number characters contained, please retry."错误信息将在输入非数字字符或回车后给出，它表明使用者输入了非法字符使得快递量不是十进制无符号非负整数。在接收到该条错误信息后需要重新输入当天快递量。

（2）"Numbers exceeded out of range, please retry."错误信息将在所输入的十进制无符号非负整数不在[0,999]区间内时给出，它表明使用者输入了不合法的快递量。在接收到该条错误信息后需要重新输入当天快递量。

（3）请注意在未输入任何字符时，按下回车键不会结束当天快递量的输入而只是光标换行。

（4）请注意输入数字字符时应使用主键盘区上方的数字键，不能使用小键盘的数字键。小键盘的数字键在 DOSBox0.74 中输入后将被显示和识别为非数字字符。

五.课程总结

1.课程总结

两周的《汇编语言程序设计》课程中，我能够按时出勤上课、上机，课后认真复习 PPT 等讲义资料，认真完成和订正练习册上的题目，及时上交作业和实验报告。虽然是网络授课，但依然能做到像在学校里一样认真听讲，能够时刻保证人在屏幕前跟随老师节奏进行学习、思考。在做课堂练习和被提问时能够很快地正确回答题目、问题。整个暑期实践期间表现很好。

虽然在这两周期间同时还有大二暑期实践课程，存在课程冲突的问题，但考虑到《汇编语言程序设计》授课内容较多且课时紧凑，因此使用一台设备全程学习《汇编语言程序设计》，另一台设备对其他课程进行录屏，等到《汇编语言程序设计》课程结束后再回看。上课时一般是手机登录 zoom 学汇编（这也就是有一次做课堂练习老师让我共享屏幕展示答案时我说我是在纸上作答的原因，电脑当时被占用无法使用），电脑挂机录屏其他课。由于下午不存在课程冲突，因此上机时可以用电脑正常进行上机。基于这种选择，在暑期实践期间完整地学习了《汇编语言程序设计》的课时内容，没有疏漏的地方。

上课时老师的节奏是比较慢的，因此讲的内容很详细、清楚，而且配合画图展示能够有效地帮助我进行理解。教学内容比较连贯，不存在知识断层的情况，PPT 环环相扣，让我比较系统、全面地学习到了汇编语言的基本内容和计算机硬件的相关知识。同时配合自己课后的复习、预习，基本上没有疑惑的地方，学习效果很好。

上机时要求开摄像头应该是特殊时期完成实践课时不得已的一种手段，我也能够理解。只是由于个人心理素质等原因，开着摄像头感觉像是随时都有人盯着自己，有一种压迫和紧张感。同时为了注意个人形象，一些比较舒服的动作（抓耳挠腮等）无法施展，导致比较拘束。所以上机时整个人处于紧绷状态，注意力不能完全集中在题目上，导致效率不是很高，往往需要关掉摄像头之后再花一些时间才能完成。

学习过程中使用的资料以老师上课时的 PPT 为主，毕竟老师第一节课时说过所讲内容基本上是够用的。我也看过网盘里的其他补充资料，和上课时所讲内容基本一致，另外讲了一些内容在本课程学习过程中使用不到，仅作为了扩展知识面使用。偶尔遇到理论上和程序设计上的问题时，会参考网络上 CSDN 社区里的一些回答进行解决。学习资料不多，但是经过充分利用之后使我受益匪浅。

对于课程知识点，基础知识和硬件基础在其他课程中有所接触，汇编课程中再次学习巩固使得掌握得非常熟练。如在《高级语言程序设计》中学过数据在内存中存储的小端模式（即高位存高地址，低位存低地址）等。数据定义部分，对于 EQU 的使用类比高级语言中的宏定义，接受起来也很容易，而对于 LABEL 的使用掌握尚有所不足。数据传送部分，由于讲课和习题很详细，关于 7 种传送方式以及注意事项了解得都非常清晰。数据运算方面，对于加减法指令掌握较好，包括双字数据高低位分别运算、双字数据求补等技巧都掌握得比较好。涉及到乘除法的，有时会忘记考虑乘除字节数据和乘除字数据的细微差异，在定义除数时将 DB 和 DW 定义错误导致运行出错，需要通过调试来进行纠正。对于运算过程中的有符号扩充、无符号扩充和溢出考虑等，通过上机题的训练已经

能够熟练掌握和使用。输入输出指令由于自己编程时几乎没有用到，因此只停留在理论层面，还未能很熟练地实践。分支程序设计时逻辑结构能够理清，做到运行正确，但在如何优化程序结构，使得多分支程序化为单分支程序方面尚有不足，JMP 使用仍然较多。对于条件跳转指令如 JA、JGE 等能够针对不同数据类型进行合理使用。循环程序设计时，能够根据不同场合熟练区分使用计数循环和条件循环，能够考虑到循环的边界情况（即是否必须循环至少一次等）。汇编语言的子程序设计与高级语言对比，传参方式不太相同，现场保护和恢复需要手动完成，其余差别并不是很大，且模块化设计的连接过程与《高级语言程序设计》中在 Linux 平台下进行多源文件程序的编译十分类似，因此也能够很好地掌握。输入输出与中断由于讲得比较快且更多地属于普及性内容，需要通过自学了解更多的系统功能才能在适当情况下进行调用，课程结束后仍然会关注相关内容。

编程能力方面，汇编语言要求我更加关注底层操作，从寄存器和内存角度理解程序的运行，因此提高了我程序设计时的严谨性。寄存器和存储器的差异，要求我在程序设计时尽量进行寄存器操作以减少内存和时间消耗等。同时用于自己能够直接修改二进制数据，使得我在编程时更多地考虑到了有符号数与无符号数的区别，以及运算是否导致溢出等问题，增强了我设计的程序的健壮性。

这门课相比我上过的许多课来说，教的算是很详细的，课程安排也很合理。建议可以适当增加课堂练习次数和难度，从而及时发现同学们问题所在，精益求精，进一步提升教学效果。

2.大作业总结

本次大作业设计中，由于假期有其他各种事情导致没有集中的时间，完成过程分散在较长的时间跨度中，累计完成时间约为 11 小时，全部在家中进行。所有设计内容均是由自己独立完成，没有和他人进行合作。

由于题目比较朴素，没有过多的额外操作需求，不需要自学新内容，只需要对分支循环程序设计和子程序设计的 PPT 进行巩固复习即可，这部分时间约占 1h。在解决跳转距离过长，超出 JA 可跳转范围的问题时参考了 CSDN 社区的相关回答。

对于小数处理，采用了计算时先同时扩大 10 倍，输出时再除以 10 分别输出商和余数的方法进行解决，这是在课上刚看到题目时就想到的，不过后来老师也给了提示。

在大作业完成过程中加深了对于乘除法运算的理解。由于程序中涉及到二进制转化为十进制输出以及小数处理等操作，因此 10 是一个重要常数，需要提前定义好。10 小于 255，理论上只需要定义成 DB 即可。然而，在进行输入时需要不停地使用乘 10 加新数字的操作，当输入类似 8500 这样超出字节数范围的数据时，如果使用 DB 定义，则 MUL 指令会将 AL 内容与 10 相乘，存于 AX 中，但实际上在这过程中 AX 会先变成 850，此时 AL 中内容不再是完整数据，与 10 相乘在存于 AX 的数据也就不是 8500，实际成为了 520。除法操作也存在类似问题。这种问题比较隐蔽，导致在 Debug 时花费了不少精力，最终由此提升了自己对于乘除法运算时各寄存器功能的认识，同时警示自己需要实现模拟计算机工作的原理、过程去设计程序，不能循规蹈矩，也不能想当然。

在大作业完成过程中加深了对提高运行速度的技巧的认识。将常量定义为 EQU，则在使用时由于是简单替换，本质仍然相当于立即数。而如果是用 DB、DW、DD 等类型进行变量定义和初始化，则在使用时需要存储器传送，从而降低了运行速度。因此，同样是为了实现使常量更有意义以及便于修改的目的，使用 EQU 是一个能够提升程序运行速度的有效技巧。

大作业全部符合题目要求，完成了所有程序功能，报告内容参照撰写说明进行了逐条的详细撰写，全部符合报告要求。程序本身尚存在不支持直接输入溢出数据（如 1000000，超过 32768）的问题，与一般的超出范围（如超出 999 给出错误提示）不同，可能会存储为一个较小但不正确的数，能够进行后续计算，但结果也不正确。为了解决这个问题可以在输入过程中判断 OF 标志位状态，对于数据过大的特殊情况报错误提示。

汇编语言由于之前没有接触过，在学习过程中很感兴趣。其理论和计算机硬件理论有很大关联，使得我能够将《计算机组成原理》相关内容与之结合进行深度思考。其实践

过程让我感受到了区别于《高级语言程序设计》的，从更底层做起的细致和严谨的要求。C++本身已经同内存等结合较多了，而汇编语言更是要求对数据存储、运算等的内存情况了如指掌。没有了高级语言方便的各种语句，而从一条条指令入手，对程序员的基础能力要求更高了。

今后我会将汇编课程所学习到的基础知识更加广泛地同后续课程内容结合起来，并且在暑假设计 54 条 MIPS 架构 CPU 时感受不同指令系统下操作的细微差别。作为计算机专业基础课程，汇编语言在跨平台方面相比 C++等高级语言无疑更有优势，观察汇编代码也可以在某些情况下对尚未掌握的高级语言代码进行 Debug 工作。因此今后我也会在遇到汇编相关问题时主动查找资料进行解决，不断扩充自己的知识领域。

附程序清单：

```
;1853971 王天，K1853971.asm
DATA SEGMENT
N     EQU 6      ;统计的天数
STA   EQU 100        ;每日基准件数
BAS    EQU 2200 ;每日基本工资
MORE EQU 15         ;超出部分每件奖励工资
LESS EQU 12         ;不足部分每件扣除工资
MIN EQU 0            ;件数下限
MAX EQU 999         ;件数上限
C10 DW 10
NUM DW N DUP(0)
ANS DD 0
FLAG DB 0  ;标志变量，记录是否读到过回车外的字符
PLEASE1 DB  "Please input the NUM of day ",'$'
PLEASE2 DB  " (range from ",'$'
PLEASE3 DB  " to ",'$'
PLEASE4 DB  ") : ",'$'
RESULT  DB "The total wage is : ",'$'
ERRMSG1 DB  0AH,0DH,"Non-number characters contained, please retry.",0AH,0DH,'$'
ERRMSG2 DB  "Numbers exceeded out of range, please retry.",0AH,0DH,'$'
DATA ENDS
CODE SEGMENT
     ASSUME  CS:CODE,DS:DATA
;-----------------------------------------------
;子程序名：DISP
;功能：将寄存器 AX 中的数据以十进制显示出来
;入口参数：AX
;出口参数：无
;-----------------------------------------------
DISP PROC FAR
    PUSH DX
    PUSH CX
    PUSH BX
    MOV CX,0    ;计数器
```

```
    MOV BX,10
REP1: MOV  DX,0
    DIV BX          ;除以 10 取余
    ADD DX,30H      ;DX 加 30H
    PUSH DX         ;先低后高入栈
    INC CX          ;计余数个数
    OR  AX,AX
    JNZ REP1        ;商 0 结束循环
REP2: POP DX        ;先高后低弹出
    MOV AH,2        ;显示
    INT 21H
    LOOP REP2
    POP BX
    POP CX
    POP DX
    RET
DISP ENDP
;-----------------------------------------------
;子程序名：INPUT
;功能：从键盘输入一周中每天的快递量并进行错误处理
;入口参数：无
;出口参数：NUM
;-----------------------------------------------
INPUT PROC FAR
    LEA BX,NUM
    MOV CX,1
L1: LEA DX,PLEASE1
    MOV AH,9
    INT 21H
    MOV AX,CX
    CALL DISP
    LEA DX,PLEASE2
    MOV AH,9
    INT 21H
    MOV AX,MIN
    CALL DISP
    LEA DX,PLEASE3
    MOV AH,9
    INT 21H
    MOV AX,MAX
    CALL   DISP
    LEA DX,PLEASE4
    MOV AH,9
    INT 21H
L2: MOV AH,1
    INT 21H
    CMP AL,0DH
    JE  DONE
```

```
    CMP AL,39H
    JA  ERROR1
    CMP AL,30H
    JB  ERROR1
    MOV FLAG,1
    MOV DX,AX   ;输入字符转移到CX寄存器
    AND DX,000FH;转换成二进制数
    MOV AX,[BX]
    PUSH DX
    MUL C10
    POP DX
    ADD AX,DX   ;新输入数字拼接到已输入数字中
    MOV [BX],AX
    JMP L2
DONE: CMP FLAG,0
    JE  L2
    CMP WORD PTR [BX],MAX
    JA  ERROR2
    CMP WORD PTR [BX],MIN
    JB  ERROR2
    INC BX
    INC BX
    INC CX
    CMP CX,N
    JA  EXIT
    MOV FLAG,0
    JMP L1
ERROR1: LEA DX,ERRMSG1
    MOV AH,9
    INT 21H
    MOV WORD PTR [BX],0
    MOV FLAG,0
    JMP L1
ERROR2: LEA DX,ERRMSG2
    MOV AH,9
    INT 21H
    MOV WORD PTR [BX],0
    MOV FLAG,0
    JMP L1
EXIT:  RET
INPUT  ENDP
;-----------------------------------------------
;子程序名：SOLVE
;功能：根据各日件数计算总工资
;入口参数：BX
;出口参数：ANS
;-----------------------------------------------
SOLVE PROC
```

```
    LEA BX,NUM
    MOV CX,N
L3: MOV AX,[BX]
    CMP AX,STA
    JA  EL
    MOV AX,STA
    SUB AX,[BX]
    MOV DX,LESS
    MUL DX
    NEG AX
    JMP AC
EL: SUB AX,STA
    MOV DX,MORE
    MUL DX
AC: ADD AX,BAS
    ADD WORD PTR [ANS],AX
    ADC WORD PTR [ANS+2],0
    INC BX
    INC BX
    LOOP L3
    RET
SOLVE ENDP
;-----------------------------------------------
;子程序名：OUTPUT
;功能：输出总工资，保留一位小数
;入口参数：ANS
;出口参数：无
;-----------------------------------------------
OUTPUT PROC
    LEA DX,RESULT
    MOV AH,9
    INT 21H
    MOV AX,WORD PTR [ANS]
    MOV DX,WORD PTR [ANS+2]
    DIV C10
    PUSH DX
    CALL DISP
    MOV AH,2
    MOV DL,'.'
    INT 21H
    POP AX
    CALL DISP
    RET
OUTPUT ENDP
;-----------------------------------------------
;子程序名：MAIN
;功能：主程序
;入口参数：无
```

```
;出口参数：无
;-----------------------------------------------
MAIN PROC  FAR
    PUSH DS
    MOV AX,0
    PUSH AX
    MOV AX,DATA
    MOV DS,AX
    CALL INPUT
    CALL SOLVE
    CALL OUTPUT
    RET
MAIN ENDP
CODE ENDS
     END MAIN
```