

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。

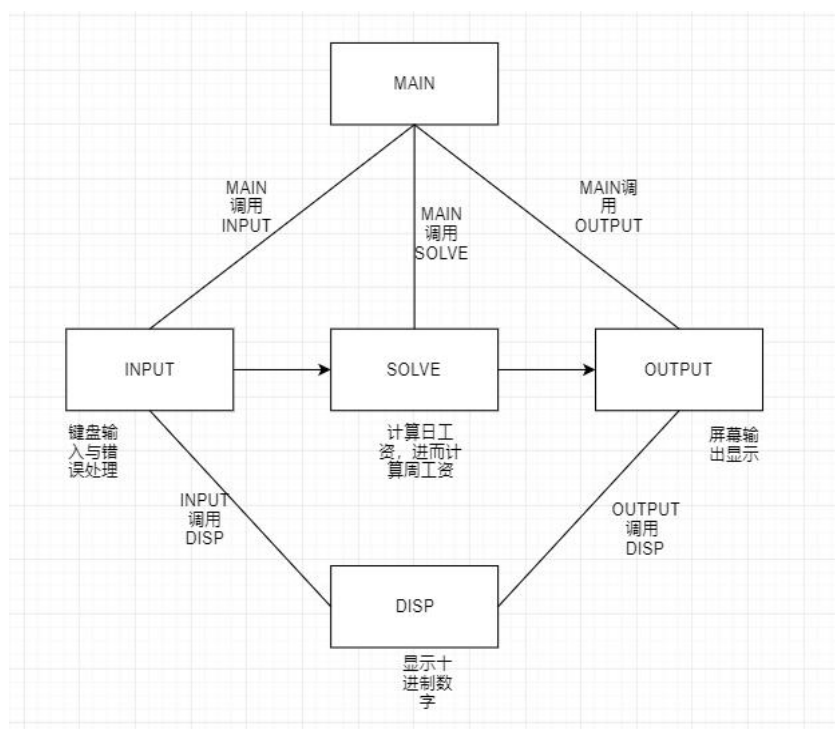
(2) INPUT：给出输入提示信息，完成数据从键盘的输入，判断输入数据的合理性并对错误数据给出错误信息。无入口参数，出口参数为 NUM(存储每天快递量)，即将所输入数据存储于 NUM 数组中。

(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。

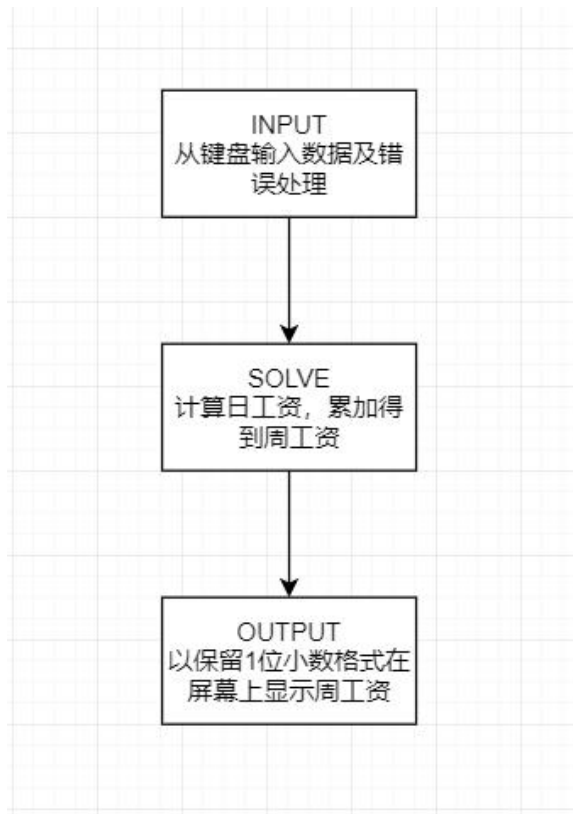
4. 程序框图如下：



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，得到每天的快递量，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 的情况下进行每天的快递量输入，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。

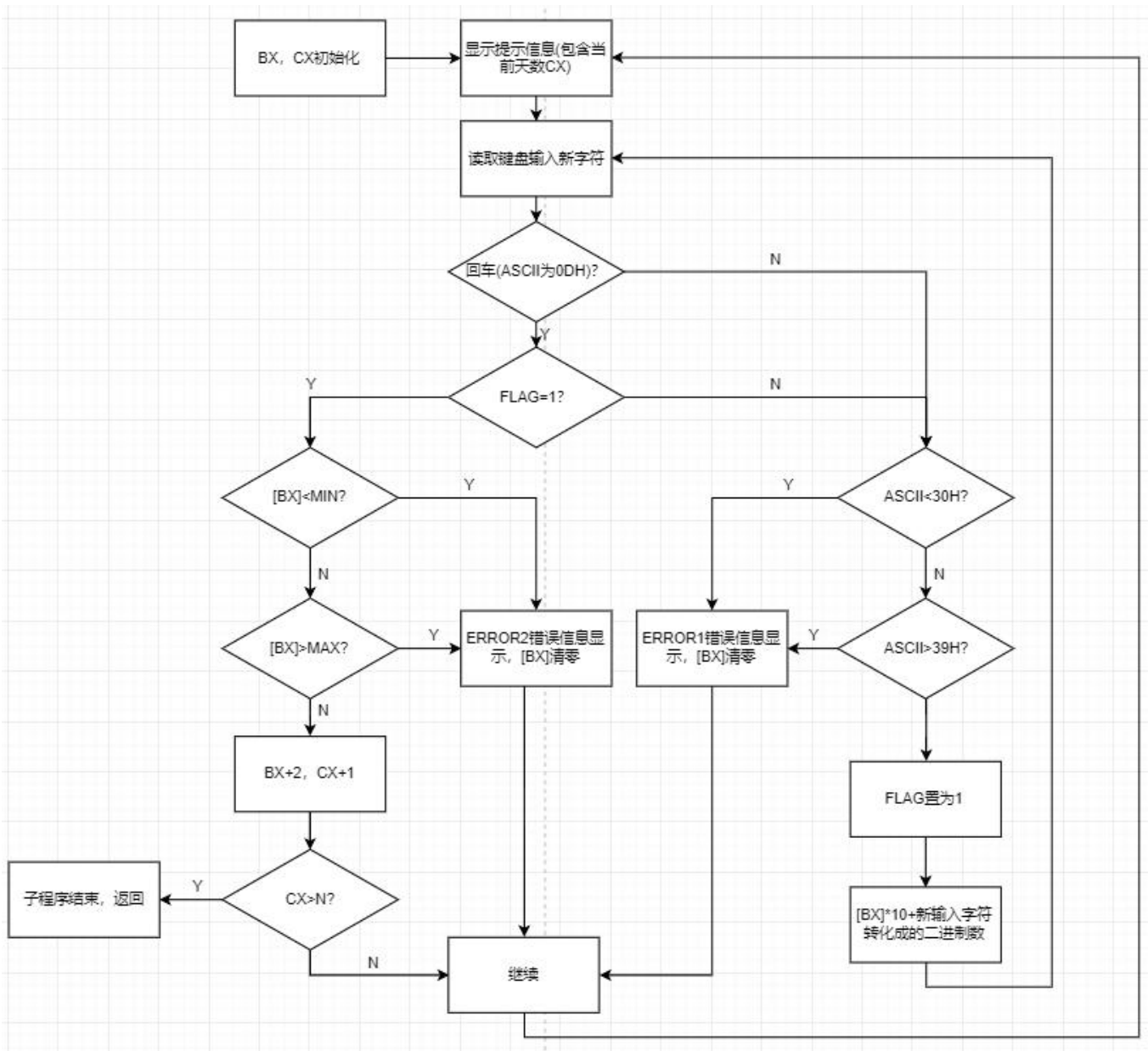
读取到回车字符时进行判断：若 FLAG 变量的值为 1，则此前读取过数字字符，结束读取；若 FLAG 变量的值为 0，则此前未读取过数字字符，继续读取。

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：



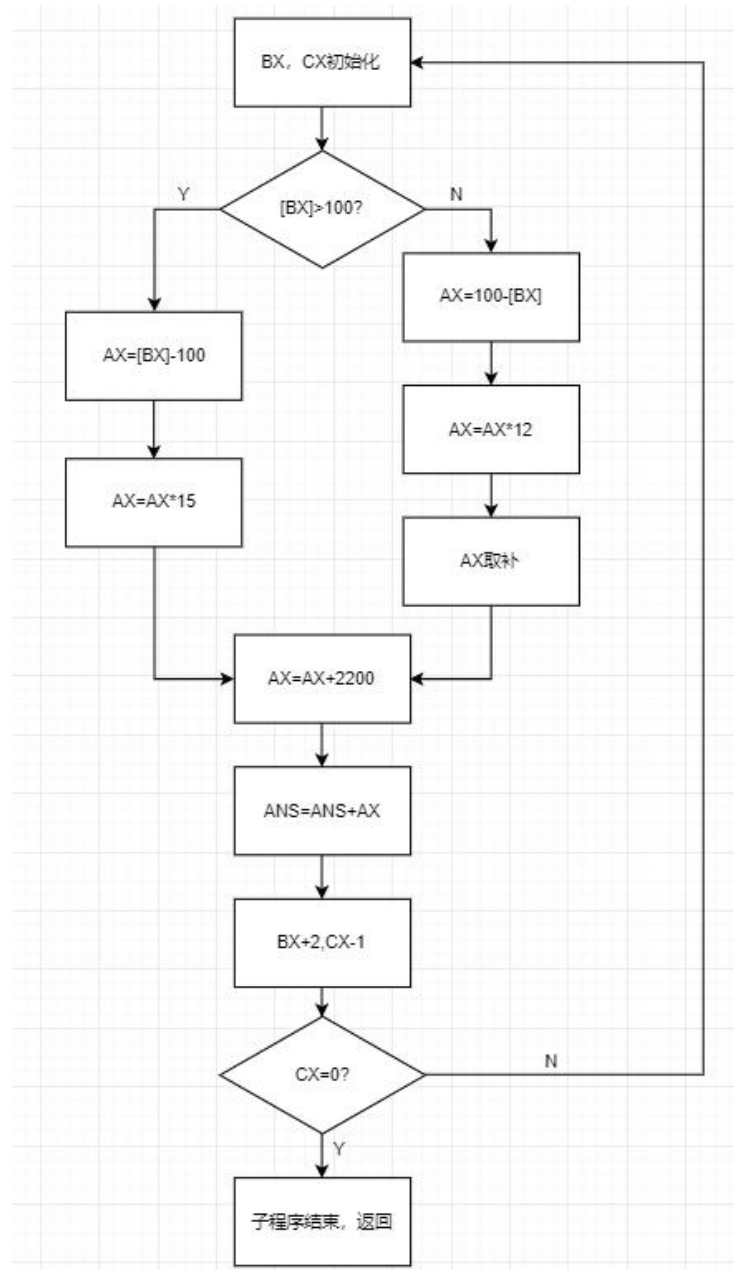
(3) SOLVE: 计算部分子程序。首先将 NUM 数组的首地址送入 BX 寄存器, 使得 BX 起到指示当前数组元素位置的作用, 将 CX 寄存器预置为 N, 之后利用计数循环 LOOP, 进行各天的工资的计算并累加, 当 CX 变为 0 时则子程序调用结束并返回。

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。

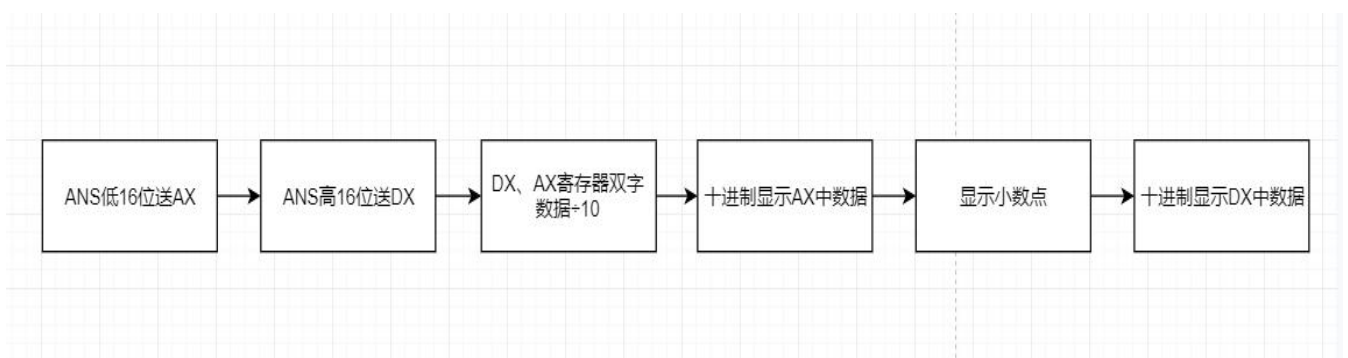
实际操作时为避免浮点数运算, 采取将数据扩大 10 倍的方法, 实际是分别乘以 15 和 12、加上 2200。同时由于扩大了 10 倍进行计算, 虽然日工资不会超过字类型有符号整数范围, 但累加结果可能会超过, 因此 ANS 需要使用双字类型进行存储(进行加法运算时也要高低位分别运算)。

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。

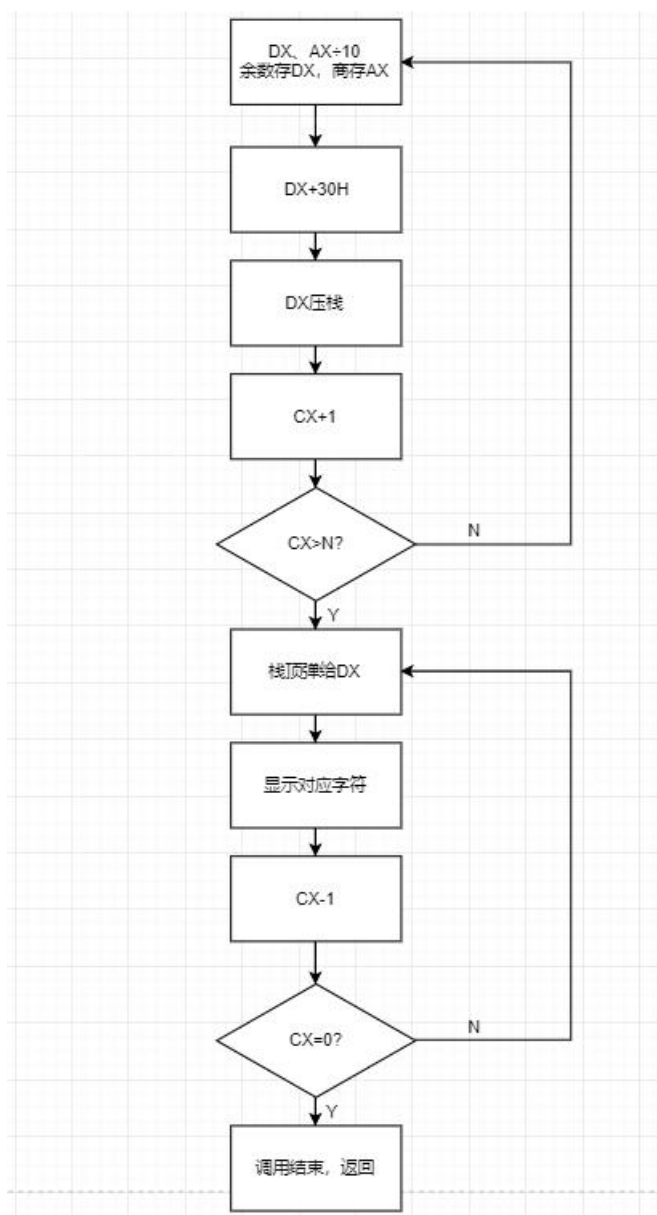
流程图如下所示:



(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，依次输入各天快速量。

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。