

第三次作业

任俊屹, PB16070892, [github](#)

1 设计一个函数子程序将一个数字型字符转化为一个与其相同的数值型数据，并调用该子程序完成：

(1) 将字符型数据 '23456' 转化为整型数据 23456

(2) 将字符型数据 '75.8' 转化为实型数据 75.8.

依次自高到低读取字符，转换成相应整数，并加入到计数器中。每读一位，计数器乘十。当读到小数点后，每读一位，小数计位乘十。最终，计数器中的值除以小数计位中的值即可。代码如下：

[1.f90](#)

```
1 subroutine Str2num(str, num)
2   implicit none
3   Character (len=*) :: str
4   Real num
5   Integer :: inte=0, dicimal=1, i
6   Logical :: dic_encountered = .false.
7
8   !Tranverse the string, get the number as float dicimal number.
9   do i = 1, len(str)
10      if ('0' < str(i:i) .and. '9' > str(i:i)) then
11         inte = inte*10 + iachar(str(i:i)) - iachar('0')
12         if (dic_encountered) then
13            dicimal = dicimal * 10
14         end if
15      else if (str(i:i) == '.') then
16         dic_encountered = .true.
17      end if
18   end do
19
20   !Choose to return Real or Integer,
21   !but I don't know if fortran supports templete.
22   if (dic_encountered) then
23      num = Real(inte) / dicimal
24   else
25      num = inte
26   end if
```

```

27 end subroutine Str2num
28
29 program main
30     implicit none
31     !Integer :: num1;
32     Real :: num1, num2;
33     Character (len=*), parameter :: str1='23456', str2='75.8'
34
35     call Str2num(str1, num1)
36     call Str2num(str2, num2)
37
38     write(*, *) num1
39     write(*, *) num2
40 end program

```

无输入文件，输出文件为[out.txt](#)。

- 2 对于任意的二维数组 $A(m,n)$ ，设计一个子例行程序 $\max(A,B,m,n,k)$ 。其中 A 是一个二维数组， m 、 n 分别是 A 的行数和列数， B 是一个一维数组。子程序的功能是：当参数 $k=1$ 时，求 A 的每列上的最大元素并存放到 $B(1)$ 、 $B(2)\dots B(n)$ 中；当参数 $k=2$ 时，求 A 每行上的最大元素并存放到 $B(1)$ 、 $B(2)\dots B(m)$ 中。

直接调用内置函数即可。代码如下：

[2.f90](#)

```

1 subroutine max(A, B, m, n, k)
2     implicit none
3     Real :: A(:, :), B(:)
4     Integer :: m, n, k
5
6     !The question looks like a description of this fuction.
7     B = maxval(A, k)
8 end subroutine max
9
10 program main
11     Real, allocatable :: A(:, :), B(:)
12     Integer :: m, n, k
13
14     interface
15         subroutine max(A, B, m, n, k)
16             Real :: A(:, :), B(:)
17             Integer :: m, n, k
18         end subroutine max

```

```

19     end interface
20
21     !Input and output, blablalba...
22     write(0, *) "Input m, n, k"
23     read(*, *) m, n, k
24     allocate(A(m, n))
25     if (k == 1) then
26         allocate(B(n))
27     else
28         allocate(B(m))
29     end if
30
31     write(0, *) "Input A"
32     read(*, *) A
33
34     call max(A, B, m, n, k)
35
36     write(*, *) B
37 end program

```

输入文件为[in.txt](#)，输出文件为[out.txt](#)，输入提示信息打印在错误流，见[err.txt](#)。

3 设计一个子例程序，将一个字符型数据翻译成密文，翻译规则是：

- (1) 当对应字符是一个英文字母时，将 A→Z(a→z), B→Y, C→X.....
- (2) 当对应字符非英文字母时，保留该字符原文。如将字符型数据 'word!' 应翻译为 'dliw!'

遍历分类处理。代码如下：

[3.f90](#)

```

1 subroutine encrypt(str)
2     Character (len=*) :: str
3     Integer :: i
4
5     !Just do what is asked.
6     do i = 0, len(str)
7         if (str(i:i) >= 'A' .and. str(i:i) <= 'Z') then
8             str(i:i) = achar(iachar('Z') - iachar(str(i:i)) + iachar('A'))
9         else if (str(i:i) >= 'a' .and. str(i:i) <= 'z') then
10             str(i:i) = achar(iachar('z') - iachar(str(i:i)) + iachar('a'))
11         end if
12     end do
13 end subroutine encrypt
14

```

```
15 program main
16   Character (len=:), allocatable :: str
17   Integer :: l
18
19   write(0, *) "Input max string length"
20   read(*, *) l
21
22   allocate(Character (len=l) :: str)
23
24   write(0, *) "Input the string"
25   read(*, *) str
26
27   !Remove the extra length of the string.
28   str = trim(str)
29   call encrypt(str)
30
31   write(*, *) str
32 end program
```

输入文件为[in.txt](#)，输出文件为[out.txt](#)，输入提示信息打印在错误流，见[err.txt](#)。