

# Math 1MP3 Midterm Test #2

## Sample Test

1. For each of the following expressions, determine if it evaluates to **True**, **False**, or it produces an **Error**. Write your answer in the box below the expression.

In the following,

```
import numpy as np
```

- `x = [0,1,2,3,4]`
- `a = np.array(range(5),float)`
- `b = np.array(x,float)`
- `c = np.arange(5, dtype = float)`
- `d = np.arange(0,5,1,dtype = float)`
- `e = np.array([[1,2,3],[4,5,6],[7,8,9]])`

Useful tip:

`np.array_equal(arr1,arr2)` will return **True** if and only if `arr1` and `arr2` are the same, that is, **both shape and element** of two arrays are the same.

Each part of this question is worth 1 point, for a **maximum of 5 points**. No partial credit will be given and no points will be deducted for an incorrect answer.

In [9]:

```
1 import numpy as np
2 x = [0,1,2,3,4]
3 a = np.array(range(5),float)
4 b = np.array(x,float)
5 c = np.arange(5, dtype = float)
6 d = np.arange(0,5,1,dtype = float)
7 e = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

### Expression 1

```
- -4**2**-1 != 1.0/16
```

#grouping order, precedence

True

在数学中 $x^{y^z}$ 等于 $x^{(y^z)}$ , 所以exponent应先从右向左计算  
先计算 $2^{-1}$ , 得出 $4^{(1/2)}$ . 2不等于  $(1.0/16)$

## Expression 2

```
9 - 4 ** 3 / 5 // 2 % 15 == 3.0
```

#grouping order, precedence, floating point number

True

在python中 `**` 有很高的优先度(precedence),仅次于括号.  
优先计算 $4^{**}3$ 得到64, 由于四则运算中乘与除优先度高于  
加与减, 所以计算 $64/5$ 得到12.8。再由 $12.8//2$ 得到6.0  
 $9-6.0 = 3.0$

## Expression 3

```
np.array_equal(a,d) and np.array_equal(b,c)
```

#array creation

True

`range(5)`与`range(0,5)`等价, 5不会被包含(本题中为从0至4的array). `np.arange(0, 5, 1)`与`np.arange(5)`  
均为从0至4创建一个array。括号中的`dtype=float`将array中创建的数字的type限定为float, 本题中不影响结果  
说人话: 上面abcd四种表达都是一个意思

## Expression 4

```
e[1,0] == x[-1]
```

#multi-dimension array, index

True

`e[1,0]`为e中第二行的第一个数字, 4.  
`x[-1]`为list x中倒数第一个数字, 4.  
 $4=4$ .  
Q.E.D

## Expression 5

```
np.array_equal(b == d ,np.array([True,True,True,True,True]))
```

#logical arrays

True

根据题干描述，np.array\_equal(arr1,arr2) return True如果arr1和arr2相同

因为b与d完全相同，而比较array时会将array中每一个element单独比较，所以b==d会

return一个由5个True组成的array，括号中的与后者完全相同

2. For each of the following python statements determine what is printed by them and write your answer in the box below the statement. If the statement or statements result in an error, your answer should be **Error**.

In the following,

- str\_1 = "I love "
- str\_2 = "python programming"
- list\_1 = [1,2,3]
- set\_1 = {4,5,6}
- tup\_1 = ("x", "y", "z")
- dict\_1 = {"java": 2000, "haskell":1000, "c":{"c#":500, "c++":300}}

Each part of this question is worth 1 point, for a **maximum of 10 points**. No partial credit will be given and no points will be deducted for an incorrect answer.

### Statement 1:

```
b = list_1
b[2] = 0
print(list_1+b)
```

#aliasing, list concatenation

[1,2,0,1,2,0]

python中创建list\_1,再令b等于list\_1，虽然list\_1和b看上去是两个不同的变量名，但他们其实都指向同一个list对任何一个做更改都会影响到另一个。(array也是相同的道理)

Statement 2

```
a = tup_1[1]
b = a
a = list_1[1]
print(b)
```

#value assignment, index

"y"

a = tup\_1[1] 得到“y”  
b=a 令b=“y”  
a = list\_1[1] 令a=2, 在本题中不影响b的值:  
1.int和string是immutable不受aliasing影响  
2.此处给a重新赋值, 并没有改变b的值的可能

Statement 3

```
str_3 = str_1[2:]
str_4 = str_2[7:-4]
print(str_3+str_4)
```

#slicing, string concatenation

“love program”

str\_1[2:]代表从第三个开始最后的所有character, 从0开始数, 包含2。也就是“love ”  
str\_2[7:-4]代表从第八个开始之后, 到-4为index为止的所有character, 从0开始数, 包含7,  
不包含-4 (可以和range一起记, 都不包含后面的数).由于str\_2的第八个字母是p, 而-4就是从后往前数  
四个字母, 得到m, 因为不包含-4, 所以取前面一个的m得到”program“。两条str连在一起就是”love program“

Statement 4

```
import numpy as np
a = np.array(list_1)
print(a.shape+tup_1)
```

1	#array creation, tuple concatenation, shape
2	
3	(3, "x", "y", "z")
4	
5	a = np.array(list_1) 创建了一个1D array, 其中的三个值分别为1, 2, 3.
6	a.shape 因为是1D array (没有row的概念)。并且有三个column (三个值), 所以a.shape会得到 (3, )
7	(3, ) + ("x", "y", "z") = (3, "x", "y", "z")

## Statement 5

```
set_1.add(6)
set_1.add(7)
set_1.add(8)
print(set_1[3])
```

#set methods, property of set

ERROR

NOTE: set\_1.add(6)依然得到{4,5,6}! 因为set内不会有重复的值  
并且因为set是没有顺序之说的, 所以试图打出第四个值自然会报错

## Statement 6

```
tup_1[0] = str(1)
```

#mutability

ERROR

tuple是immutable, 一旦赋值就不可更改, 除非重新赋值。  
本题试图修改tuple中的第一个值, 导致报错

## Statement 7

```
import numpy as np
a1 = np.array(tup_1)
b1 = a1[1:3]
b1[0]="x"
print(b1)
```

#array creation, array slicing, mutability

['x' 'z']

正如list可以创建array一样, tuple也行。得到a1为["x" "y" "z"]。(注意是type为array, 不是list)  
正如string可以slice一样, array也行。a1[1:3]是从第二个值开始, 到第四个的前一个值结束  
也就是“y”和“z”, 然而本题中将b1的第一个值换成了“x”, 所以最终会print出["x" "z"], 注意是type为array, 不是list

## Statement 8

```
list_3 = []  
for i in range(1,len(list_1)):  
    list_3.append(list_1[0:3:i])  
print(list_3)
```

#range, list expansion, advanced slicing, for loop

[[1, 2, 3], [1, 3]]

len(list\_1): list\_1的长度, 3

此for loop循环两次, 由于从1开始, 3结束 (不包括3)

第一次循环: i=1, 让list\_3 append list\_1中从0到3的 (不包含3) 所有character, 一个一个数。得到[1,2,3]

第二次循环: i=2, 让list\_3 append list\_1中从0到3的 (不包含3) 所有character, 每数完一个数就跳一个。得到[1,3]

append两个list进list\_3得到[[1,2,3],[1,3]]

## Statement 9

```
for something in dict_1:  
    print(something)
```

#for loop, dictionary

“java”  
”haskell“  
”c“

for something in dict\_1:  
something只有可能是这个dictionary的key。

## Statement 10

```
list_1.append(set_1)  
list_1.append(tup_1)  
list_1.append(list_1)  
list_1.append(dict_1)  
print(len(list_1)//2)  
print(len(list_1)%2)  
print(len(list_1)/2)
```

#list methods, arithmetics

3

1

3.5

第一步：append一个set进list，得到[1,2,3,{4,5,6}] length为4

第二步：又append一个tuple进list，得到[1,2,3,{4,5,6},(4,5,6)] length为5

第三步第四步：略 反正每次都append一个整体，长度都只会增加1

7//2，得到3.0 （只要float出现在计算中，结果也一定是float）

7%2，得到1 （7除以2的余数为1）

7/2，得到3.5

3. Evaluate each of the following expressions and write the result in the box below the expression. If the expression can not be evaluated, you should write **Error** as your answer.

Each part of this question is worth 1 point, for a maximum of 3 points. No partial credit will be given and no points will be deducted for an incorrect answer.

For the following,

```
import numpy as np
```

- `a = np.array([[1, 2], [3, 4], [5, 6]], float)`
- `b = np.array([-1, 3], float)`
- `c = np.array([-5, 0], float)`

### Expression 1

```
(a+b)[2,0]
```

#array mathematics, index, broadcasting

4.0

根据定义：arrays that do not match in the number of dimensions will be broadcast (to perform mathematical operations)

I this means that the smaller array will be repeated as necessary to perform the operation.

将b变为一个shape为（3，2）的array进行计算，变换之后每一row都相同，为[-1 3]

经过计算第三row为[4.0 9.0]，结果要求第三row的第一col，答案是4.0

NOTE：括号中的float限定了array中的type，所以有小数点

## Expression 2

```
np.dot(a.flatten(),a.flatten())
```

#vector mathematics, reshape

91.0

a.flatten()是将a变成一个1D array。 [1 2 3 4 5 6]

np.dot()是计算括号中前后两个matrix(将array看成1\*6的matrix)的dot product

$1.0+4.0+9.0+16.0+25.0+36.0=91.0$

注意type为float

## Expression 3

```
(b/c)[0,0]
```

#array mathematics

ERROR

array c中包含了0, 报错

4. For each of the functions defined below, provide a **brief** description, in the space just below the code, of what it does. Do not just describe what the individual lines of code do. Each part of this question is worth a **maximum of 2 points**

**(a)** In the following, **arr** is a **sorted in ascending order** non-empty list with 6 integers, and **x** is an integer.

```
def fun_1 (arr,x, lower=0, upper=5):  
    while lower <= upper:  
        mid = lower + (upper - lower)//2;  
        if arr[mid] == x:  
            return mid  
        elif arr[mid] < x:  
            lower = mid + 1  
        else:  
            upper = mid - 1  
    return -1
```



## Binary search

首先将x与一个长度为6的List中间位置上的数字比较，如果相等，则return x所在的index  
否则，若比x小，则在List前半部分中继续寻找是否有数等于x；（由于list是从小到大排列的）  
若比x大，则在List后半部分中继续进行寻找是否有数等于x。  
每经过一次比较就缩小一半的检索区间，如此进行下去，直到找到x。  
若x未在list中出现，则return -1代表未找到x

(b) In the following, **file\_1** is a plain text file that is already open for reading.

```
def fun_2(file_1):  
    lines = file_1.readlines()  
    for i in range(1,len(lines),2):  
        for word in lines[i].split():  
            print(word.upper())
```

读取文件，储存至lines中

从第二行开始到这个文件的最后，只读取偶数行，将读取到的行中的每一个word按空格分割开，并且转换成大写print出来(print出来是没有空格的)

(c) In the following, **arr** is an array of integers.

```
import numpy as np  
def fun_3(arr, change_value, new_value):  
    new_arr = arr.copy()  
    change_positions = new_arr == change_value  
    new_arr[change_positions] = new_value  
    return new_arr
```

创建原array的copy，将新array中所有等于change\_value的数变为新的数：new\_value，而不改变原array的值。（避免aliasing的影响）

5. **Each part of this question is worth 3 points.** Partial credits will be given as appropriate.

(a) Circle or indicate where all of the mistakes/omissions/errors in syntax and logic are in the code. In the box below the code, write a corrected version of the code.

The following function has one integer argument **n** and returns the sum of the first n terms of the harmonic series. If n is less than 1, then the function should return the value 0.0. So, sum harm(3) should return the value 1.8333333333333333, which is  $1/1 + 1/2 + 1/3$ .

```
def sum_harm():
    if n < 1:
        return 0.0
    else:
        sum = 0
        for counter in range(1,n):
            sum += 1//counter
        return n
```

In [ ]:

```
1
2 def sum_harm():                                <----- missing parameter n
3     if n < 1:
4     return 0.0                                <----- improper indent
5     else:
6         sum = 0
7         for counter in range(1,n):            <----- n should be n + 1
8             sum += 1//counter                  <----- // should be /
9     return n                                  <----- should return sum
10
11 Correct version:
12 def sum_harm(n):
13     if n < 1:
14         return 0.0
15     else:
16         sum = 0
17         for counter in range(1,n+1):
18             sum += 1/counter
19     return sum
```

(b) In the box below write code for a function **scalar\_product(file1,file2)** that has two arguments: **file1** and **file2**, both both have an unknown number of lines, each line consisting of a single positive integer. The function reads a line from one file and then a line from the other file. The two integers are multiplied together and their product is added to a variable called **scalar\_product** which should be initialized to zero.

Your code should stop when it detects end of file in either file that it is reading. When it stops, it should return **scalar\_product**.

For example, if the sequence of integers in one file was "9 7 5 18 13 2 22 16" and "4 7 8 2" in the other file, your code would compute:

$$4 \times 9 + 7 \times 7 + 8 \times 5 + 2 \times 18$$

and thus return 161.

In [5]:

```
1  def scalar_product(file1,file2):
2      #初始化
3      scalar_product=0
4
5      #分别打开两个文件
6      f1=open(file1,"r")
7      f2=open(file2,"r")
8
9      #分别读取两个文件
10     file1_content=f1.readlines()
11     file2_content=f2.readlines()
12
13     #循环次数为两个文件list长度的最小值
14     times = min(len(file1_content),len(file2_content))
15
16     #将结果计算, 并存入list中
17     for i in range(0, times):
18         scalar_product += file1_content[i]*file2_content[i]
19
20     #关闭文件
21     f1.close()
22     f2.close()
23
24     return scalar_product
```

# Math 1MP3 Midterm Test #2

## Sample Test

1. For each of the following expressions, determine if it evaluates to **True**, **False**, or it produces an **Error**. Write your answer in the box below the expression.

In the following,

```
import numpy as np
```

- `x = [0,1,2,3,4]`
- `a = np.array(range(5),float)`
- `b = np.array(x,float)`
- `c = np.arange(5, dtype = float)`
- `d = np.arange(0,5,1,dtype = float)`
- `e = np.array([[1,2,3],[4,5,6],[7,8,9]])`

Useful tip:

`np.array_equal(arr1,arr2)` will return **True** if and only if `arr1` and `arr2` are the same, that is, **both shape and element** of two arrays are the same.

Each part of this question is worth 1 point, for a **maximum of 5 points**. No partial credit will be given and no points will be deducted for an incorrect answer.

In [9]:

### Expression 1

```
- 4**2**-1 != 1.0/16
```

In [5]:

Out[5]:

True

### Expression 2

```
9 - 4 ** 3 / 5 // 2 % 15 == 3.0
```

In [6]:

Out[6]:

True

### Expression 3

```
np.array_equal(a,d) and np.array_equal(b,c)
```

In [10]:

Out[10]:

True

### Expression 4

```
e[1,0] == x[-1]
```

In [11]:

Out[11]:

True

### Expression 5

```
np.array_equal(b == d ,np.array([True,True,True,True,True]))
```

In [12]:

Out[12]:

True

2. For each of the following python statements determine what is printed by them and write your answer in the box below the statement. If the statement or statements result in an error, your answer should be **Error**.

In the following,

- `str_1 = "I love "`
- `str_2 = "python programming"`
- `list_1 = [1,2,3]`
- `set_1 = {4,5,6}`
- `tup_1 = ("x", "y", "z")`
- `dict_1 = {"java": 2000, "haskell":1000, "c":{"c#":500, "c++":300}}`

Each part of this question is worth 1 point, for a **maximum of 10 points**. No partial credit will be given and no points will be deducted for an incorrect answer.

In [23]:

**Statement 1:**

```
b = list_1
b[2] = 0
print(list_1+b)
```

In [14]:

```
[1, 2, 0, 1, 2, 0]
```

**Statement 2**

```
a = tup_1[1]
b = a
a = list_1[1]
print(b)
```

In [16]:

```
y
```

**Statement 3**

```
str_3 = str_1[2:]
str_4 = str_2[7:-4]
print(str_3+str_4)
```

In [17]:

```
love program
```

#### Statement 4

```
import numpy as np
a = np.array(list_1)
print(a.shape+tup_1)
```

In [18]:

```
(3, 'x', 'y', 'z')
```

#### Statement 5

```
set_1.add(6)
set_1.add(7)
set_1.add(8)
print(set_1[3])
```

In [19]:

```
-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-19-8d5189bd9e08> in <module>()
      3 set_1.add(7)
      4 set_1.add(8)
----> 5 print(set_1[3])
```

**TypeError:** 'set' object does not support indexing

#### Statement 6

```
tup_1[0] = str(1)
```

In [20]:

```
-----  
-----  
TypeError                                Traceback (most recent call  
last)
```

```
<ipython-input-20-88a7d42910f8> in <module>()  
      1 # mutability  
----> 2 tup_1[0] = str(1)
```

```
TypeError: 'tuple' object does not support item assignment
```

## Statement 7

```
import numpy as np  
a1 = np.array(tup_1)  
b1 = a1[1:3]  
b1[0]="x"  
print(b1)
```

In [21]:

```
['x' 'z']
```

## Statement 8

```
list_3 = []  
for i in range(1,len(list_1)):  
    list_3.append(list_1[0:3:i])  
print(list_3)
```

In [24]:

```
[[1, 2, 3], [1, 3]]
```



### Statement 9

```
for something in dict_1:
    print(something)
```

In [25]:

```
java
haskell
c
```

### Statement 10

```
list_1.append(set_1)
list_1.append(tup_1)
list_1.append(list_1)
list_1.append(dict_1)
print(len(list_1)//2)
print(len(list_1)%2)
print(len(list_1)/2)
```

In [26]:

```
3
1
3.5
```

3. Evaluate each of the following expressions and write the result in the box below the expression. If the expression can not be evaluated, you should write **Error** as your answer.

Each part of this question is worth 1 point, for a maximum of 3 points. No partial credit will be given and no points will be deducted for an incorrect answer.

For the following,

```
import numpy as np
```

- `a = np.array([[1, 2], [3, 4], [5, 6]], float)`
- `b = np.array([-1, 3], float)`
- `c = np.array([-5, 0], float)`

In [27]:

### Expression 1

```
(a+b)[2,0]
```

```
In [28]:
```

```
Out[28]:
```

```
4.0
```

### Expression 2

```
np.dot(a.flatten(),a.flatten())
```

```
In [29]:
```

```
Out[29]:
```

```
91.0
```

### Expression 3

```
(b/c)[0,0]
```

```
In [30]:
```

```
c:\users\shens12\appdata\local\programs\python\python36-32\lib\site-packages\ipykernel_launcher.py:2: RuntimeWarning: divide by zero encountered in true_divide
```

```
-----
-----
IndexError                                Traceback (most recent call
last)
<ipython-input-30-dcb3b7e43606> in <module>()
      1 # array mathematics
----> 2 (b/c)[0,0]
```

```
IndexError: too many indices for array
```

4. For each of the functions defined below, provide a **brief** description, in the space just below the code, of what it does. Do not just describe what the individual lines of code do. Each part of this question is worth a **maximum of 2 points**

(a) In the following, **arr** is a **sorted in ascending order** non-empty list with 6 integers, and **x** is an integer.

```
def fun_1 (arr,x, lower=0, upper=5):
    while lower <= upper:
        mid = lower + (upper - lower)//2;
        if arr[mid] == x:
            return mid
        elif arr[mid] < x:
            lower = mid + 1
        else:
            upper = mid - 1
    return -1
```

In [ ]:

(b) In the following, **file\_1** is a plain text file that is already open for reading.

```
def fun_2(file_1):
    lines = file_1.readlines()
    for i in range(1,len(lines),2):
        for word in lines[i].split():
            print(word.upper())
```

In [ ]:

(c) In the following, **arr** is an array of integers.

```
import numpy as np
def fun_3(arr, change_value, new_value):
    new_arr = arr.copy()
    change_positions = new_arr == change_value
    new_arr[change_positions] = new_value
    return new_arr
```

In [ ]:

5. **Each part of this question is worth 3 points.** Partial credits will be given as appropriate.

(a) Circle or indicate where all of the mistakes/omissions/errors in syntax and logic are in the code. In the box below the code, write a corrected version of the code.

The following function has one integer argument **n** and returns the sum of the first n terms of the harmonic series. If n is less than 1, then the function should return the value 0.0. So, sum harm(3) should return the value 1.8333333333333333, which is  $1/1 + 1/2 + 1/3$ .

```
def sum_harm():
    if n < 1:
        return 0.0
    else:
        sum = 0
        for counter in range(1,n):
            sum += 1//counter
        return n
```

In [ ]:

(b) In the box below write code for a function **scalar\_product(file1,file2)** that has two arguments: **file1** and **file2**, both both have an unknown number of lines, each line consisting of a single positive integer. The function reads a line from one file and then a line from the other file. The two integers are multiplied together and their product is added to a variable called **scalar\_product** which should be initialized to zero.

Your code should stop when it detects end of file in either file that it is reading. When it stops, it should return **scalar\_product**.

For example, if the sequence of integers in one file was "9 7 5 18 13 2 22 16" and "4 7 8 2" in the other file, your code would compute:

$$4 \times 9 + 7 \times 7 + 8 \times 5 + 2 \times 18$$

and thus return 161.

In [ ]: