



# Python 五级

2024 年 09 月

## 1 单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	B	C	A	C	C	B	C	C	A	D	D	C	D	D	C

第 1 题 在升序数组 nums 中寻找目标值 target，下列程序可以填入的是（ ）

```
1 class Search(object):
2     def search(self, nums, target):
3         left, right = 0, len(nums) - 1
4         while left <= right:
5             _____
6             if nums[mid] == target:
7                 return mid
8             elif nums[mid] > target:
9                 right = mid - 1
10            else:
11                left = mid + 1
12            return -1
```

- A.  $mid = (right + left) // 2 + left$
- B.  $mid = (right - left) // 2 + left$
- C.  $mid = (right - left) // 2 - right$
- D.  $mid = (right + left) // 2 - left$

第 2 题 500 个病毒样本中，已知有一个是病毒检测呈阳性，用试纸测试阳性病毒以后，试纸在 3 天以后会变色，用试纸测试时间不计，三天以后要出结果，请问最少用多少个试纸能够找出哪一个病毒样本有毒（ ）

- A. 499
- B. 250
- C. 9
- D. 125

第 3 题 一名收银员，给顾客找零，找零的目标是给出确定金额的同时，使用尽可能少的硬币。有不同面额的硬币：1 分，5 分，10 分，25 分。如果需要给顾客准确的零钱 77 分，同时使用最少的硬币下列程序中横线应该填写（ ）。

```

1 def coin_change(amount, coins):
2     result = []
3     for coin in sorted(coins, reverse=True):
4         while amount >= coin:
5             _____
6                 result.append(coin)
7     return result
8
9 coins = [1, 5, 10, 25]
10 amount = 63

```

- A. amount -= coin
- B. amount <= coin
- C. amount >= coin
- D. amount += coin

第4题 下列程序是素数筛的程序，横线处应该填上（ ）。

```

1 def sieve(n):
2     if n < 2:
3         return []
4     prime = [True] * (n+1)
5     prime[0] = prime[1] = False
6     for i in range(2, int(math.sqrt(n)) + 1):
7         if prime[i]:
8             _____
9                 prime[j] = False
10    return [p for p in range(2, n+1) if prime[p]]
11 for prime in sieve_of_eratosthenes(100):
12    print(prime)

```

- A. for j in range(i, n+1, i):
- B. for j in range(i\*i, 1, n):
- C. for j in range(i\*i, n+1, i):
- D. for j in range(i, n, i):

第5题 下面程序是埃氏筛的一个实现，横线处应该填写（ ）。

```

1 n = 10**8
2 s = [0]*(n+1)
3 k=0
4 for i in range(2,n+1):
5     if s[i]==0:
6         k+=1
7         _____
8             s[j]=1

```

- A. for i in range(i\*i,n+1,i):
- B. for j in range(i\*i,n,j):

C. `for j in range(i*i,n+1,i):`

D. `for j in range(j*j,n+1,i):`

第6题 下列程序中，使用了二分查找算法，横线处应该填写的是（）。

```
1 def search(arr, x):
2     low = 0
3     high = len(arr) - 1
4     while low <= high:
5         _____
6         if arr[mid] == x:
7             return mid
8         elif arr[mid] > x:
9             high = mid - 1
10        else:
11            low = mid + 1
12    return -1
```

A. `mid = (low - high) // 2`

B. `mid = (low + high) // 2`

C. `mid = (low + high) / 2`

D. `mid = (low - high) / 2`

第7题 正整数1024的所有约数的和为多少( )。

A. 2050

B. 2059

C. 2047

D. 2044

第8题 下面程序是对n! 进行唯一分解，横线处应该填入的是（）。

```
1 def unique_fac(n):
2     print(n, '=', end='')
3     for i in range(2, n + 1):
4         _____
5         print(' {}*'.format(i), end='')
6         n //= i
7         if n % i == 0 and i == n:
8             print(' {}'.format(i), end='')
9             break
10
11 unique_fac(math.factorial(5))
```

A. `while n % i != 0 and i != n:`

B. `while n % i == 0 and i == n:`

C. `while n % i == 0 and i != n:`

D. while n % i != 0 and i == n:

第9题 假设有一些物品，每个物品都有自己的重量，我们需要将这些物品装入箱子中，每个箱子也有自己的重量限制。贪心算法每次都选择重量最轻的物品放入当前最轻的箱子中，如果箱子可以装下，就放入；如果箱子不能装下，就尝试下一个箱子，直到找到可以放入的箱子。下列贪心算法程序中，横线处应该填入的是（ ）。

```
1 def box_packing(items, boxes):
2     boxes.sort(key=lambda x: x[0])
3     items.sort()
4     taken = [False] * len(items)
5     for i, item in enumerate(items):
6         taken[i] = True
7         for j, box in enumerate(boxes):
8             if box[0] >= item:
9                 _____
10                break
11     return [(box[1], sum(taken)) for box in boxes]
```

A. boxes[j] = (box[0] - item, boxes[j][1])

B. boxes[i] = (box[0] - item, boxes[j][1])

C. boxes[j] = (box[0] - item, boxes[i][1])

D. boxes[i] = (box[0] - item, boxes[i][1])

第10题 下列归并算法程序中，横线处应该填入的是（ ）。

```
1 def merge_sort(arr):
2     if len(arr) <= 1:
3         return arr
4     mid = len(arr) // 2
5     left = arr[:mid]
6     right = arr[mid:]
7     merge_sort(left)
8     merge_sort(right)
9     return merge(left, right)
10
11 def merge(left, right):
12     result = []
13     i, j = 0, 0
14     _____
15     if left[i] < right[j]:
16         result.append(left[i])
17         i += 1
18     else:
19         result.append(right[j])
20         j += 1
21     result += left[i:]
22     result += right[j:]
23     return result
```

A. while i > len(left) and j < len(right):

B. while i < len(left) and j > len(right):

C. while i > len(left) and j > len(right):

D. while i < len(left) and j < len(right):

第 11 题 下列快速排序算法中，横线处应该填入的是（ ）。

```
1 def quick(arr):
2     if len(arr) <= 1:
3         return arr
4     _____
5     left = [x for x in arr if x < p]
6     middle = [x for x in arr if x == p]
7     right = [x for x in arr if x > p]
8     return quick(left) + middle + quick(right)
```

A. p = arr[len() // 2]

B. p = arr[len(arr)+1 // 2]

C. p = arr[len(arr)-1 // 2]

D. p = arr[len(arr) // 2]

第 12 题 下列二分枚举算法中，{}处应该填入的程序是（{}不算做程序的一部分）（ ）。

```
1 def binary_search(arr, x):
2     low = 0
3     high = len(arr) - 1
4     while low <= high:
5         {
6
7     }
8     return -1
```

A. ```Python

```
mid = (low + high) // 2
```

```
if arr[mid] == x:
```

```
    return mid
```

```
elif arr[mid+1] > x:
```

```
    high = mid - 1
```

```
else:
```

```
    low = mid + 1
```

```
1
2 - [ ] **B.** ```Python
3 mid = (low + high) // 2
4     if arr[mid] != x:
5         return mid
6     elif arr[mid+1] > x:
7         high = mid - 1
8     else:
9         low = mid + 1
```

```
 C. ```Python
mid = (low + high) // 2
if arr[mid] == x:
    return mid
elif arr[mid] > x:
    high = mid - 1
else:
    low = mid + 1
```

```
1 |
2 | - [ ] **D.** ```Python
3 | mid = (low + high) // 2
4 |     if arr[mid] != x:
5 |         return mid
6 |     elif arr[mid] > x:
7 |         high = mid - 1
8 |     else:
9 |         low = mid + 1
```

第13题 下面代码是寻找水仙花数的程序，横线处应该填写的代码是（ ）。【是指一个n位数（n≥3），其每位数字的n次幂之和等于它本身】

```
1 | def is_narcissistic_num(num):
2 |     str_num = str(num)
3 |     num_digits = len(str_num)
4 |     _____
5 |     return num == sum_of_powers
6 |
7 | for i in range(100, 10000):
8 |     if is_narcissistic_num(i):
9 |         print(i, "是水仙花数")
```

```
 A. ```Python
sum_of_powers = sum(int(digit) ** num_digits for digit in num)
```

```
1 |
2 | - [ ] **B.** ```Python
3 | sum_of_powers = sum(int(digit) ** num for digit in str_num)
```

```
 C. ```Python
sum_of_powers = sum(int(num) ** num_digits for digit in str_num)
```

```
1 |
2 | - [ ] **D.** ```Python
3 | sum_of_powers = sum(int(digit) ** num_digits for digit in str_num)
```

第14题 对于正整数n，欧拉函数f(n),表示小于或等于n的正整数中与n互质的数的数目，例如f(8)=4。f(100)=（ ）。

- A. 50
- B. 55
- C. 45

D. 40

第 15 题 下列程序输出的是()。

```
1 def reverse(string):
2     if len(string) == 0:
3         return
4     temp = string[0]
5     reverse(string[1:])
6     print(temp, end='')
7 string = "chen a dai"
8 reverse(string)
```

- A. chen a dai
- B. dai a chen
- C. iad a nehc
- D. chenadai

## 2 判断题 (每题 2 分, 共 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	√	√	√	×	√	√	√	√	√	×

第 1 题  $(-1) \bmod 127$ 和 $126 \bmod 127$  的结果是一样的

第 2 题 一个数的反码, 实际上是这个数对于一个模的同余数

第 3 题 1997和615用欧几里得算法计算最大公约数的过程如下:

```
1 1997/615=3(余152)
2 615/152=4(余7)
3 152/7=21(余5)
4 7/5=1(余2)
5 5/2=2(余1)
6 2/1=2(余0)
7 得出最大公约数是1
```

第 4 题 欧几里得算法适用于实数

第 5 题 每个大于1的整数可以唯一地写成质数的乘积的形式

第 6 题 贪婪算法的复杂度通常是线性的, 即 $O(n)$ , 其中 $n$ 是输入的大小

第 7 题 归并排序的时间复杂度为 $O(n \log n)$

第 8 题 根据同余计算, 可以推导出 $(a*b)\%m=(a\%m*b\%m)\%m$

第 9 题 二分查找算法的复杂度通常表示为 $O(\log n)$ , 其中 $n$ 是数组的长度

第 10 题  $\text{def}(\text{十六进制}) = 103231(\text{五进制})$

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：小杨的武器
- 时间限制：1.0 s
- 内存限制：512.0 MB

##### 3.1.1 题面描述

小杨有  $n$  种不同的武器，他对第  $i$  种武器的初始熟练度为  $c_i$ 。

小杨会依次参加  $m$  场战斗，每场战斗小杨只能且必须选择一种武器使用，假设小杨使用了第  $i$  种武器参加了第  $j$  场战斗，战斗前该武器的熟练度为  $c'_i$ ，则战斗后小杨对该武器的熟练度会变为  $c'_i + a_j$ 。需要注意的是， $a_j$  可能是正数，0 或负数，这意味着小杨参加战斗后对武器的熟练度可能会提高，也可能会不变，还有可能降低。

小杨想请你编写程序帮他计算出如何选择武器才能使得  $m$  场战斗后，自己对  $n$  种武器的熟练度的最大值尽可能大。

##### 3.1.2 输入格式

第一行包含两个正整数  $n, m$ ，含义如题面所示。

第二行包含  $n$  个正整数  $c_1, c_2, \dots, c_n$ ，代表小杨对武器的初始熟练度。

第三行包含  $m$  个正整数  $a_1, a_2, \dots, a_m$ ，代表每场战斗后武器熟练度的变化值。

##### 3.1.3 输出格式

输出一个整数，代表  $m$  场战斗后小杨对  $n$  种武器的熟练度的最大值最大是多少。

##### 3.1.4 样例1

```
1 | 2 2
2 | 9 9
3 | 1 -1
```

```
1 | 10
```

一种最优的选择方案为，第一场战斗小杨选择第一种武器，第二场战斗小杨选择第二种武器。

子任务编号	数据点占比	$n$	$m$
1	20%	$= 1$	$\leq 10^5$
2	20%	$\leq 10^5$	$= 2$
3	60%	$\leq 10^5$	$\leq 10^5$

对于全部数据，保证有  $1 \leq n, m \leq 10^5$ ， $-10^4 \leq c_i, a_i \leq 10^4$ 。

##### 3.1.5 参考程序

```
1 | N = 100010
2 |
3 | a = [0] * N
4 | c = [0] * N
5 |
6 | n, m = map(int, input().split())
```



```

7  mx = -10000
8  s = input().split()
9  for i in range(1, n + 1):
10     c[i] = int(s[i-1])
11     mx = max(mx, c[i])
12  s = input().split()
13  for i in range(1, m + 1):
14     a[i] = int(s[i-1])
15
16  for i in range(1, m + 1):
17     if n == 1 or a[i] > 0:
18         mx += a[i]
19
20  print(mx)

```

## 3.2 编程题 2

- 试题名称: 挑战怪物
- 时间限制: 1.0 s
- 内存限制: 512.0 MB

### 3.2.1 题面描述

小杨正在和一个怪物战斗，怪物的血量为  $h$ ，只有当怪物的血量恰好为 0 时小杨才能够成功击败怪物。

小杨有两种攻击怪物的方式：

- 物理攻击。假设当前为小杨第  $i$  次使用物理攻击，则会对怪物造成  $2^{i-1}$  点伤害。
- 魔法攻击。小杨选择任意一个质数  $x$  ( $x$  不能超过怪物当前血量)，对怪物造成  $x$  点伤害。由于小杨并不擅长魔法，他只能使用至多一次魔法攻击。

小杨想知道自己能否击败怪物，如果能，小杨想知道自己最少需要多少次攻击。

### 3.2.2 输入格式

第一行包含一个正整数  $t$ ，代表测试用例组数。

接下来是  $t$  组测试用例，对于每组测试用例，第一行包含一个正整数  $h$ ，代表怪物血量。

### 3.2.3 输出格式

对于每组测试用例，如果小杨能够击败怪物，输出一个整数，代表小杨需要的最少攻击次数，如果不能击败怪物，输出  $-1$ 。

### 3.2.4 样例1

```

1  3
2  6
3  188
4  9999

```

```

1  2
2  4
3  -1

```

对于第一组测试用例，一种可能的最优方案为，小杨先对怪物使用魔法攻击，选择质数 5 造成 5 点伤害，之后对怪物使用第 1 次物理攻击，造成  $2^{1-1} = 1$  点伤害，怪物血量恰好为 0，小杨成功击败怪物。

子任务编号	数据点占比	$t$	$h$
1	20%	$\leq 5$	$\leq 10$
2	20%	$\leq 10$	$\leq 100$
3	60%	$\leq 10$	$\leq 10^5$

对于全部数据，保证有  $1 \leq t \leq 10, 1 \leq h \leq 10^5$ 。

### 3.2.5 参考程序

```
1 prime = []
2 is_prime = [False] * 100010
3
4 def Eratosthenes(n):
5     is_prime[0] = is_prime[1] = False
6     for i in range(2, n + 1):
7         is_prime[i] = True
8     for i in range(2, n + 1):
9         if is_prime[i]:
10            prime.append(i)
11            if i * i > n:
12                continue
13            for j in range(i * i, n + 1, i):
14                is_prime[j] = False
15
16 Eratosthenes(100000)
17
18 t = int(input())
19 for _ in range(t):
20     tmp = 1
21     x = int(input())
22     ans = 0
23     while True:
24         if is_prime[x]:
25             ans += 1
26             break
27         x -= tmp
28         ans += 1
29         if x <= 0:
30             if x < 0:
31                 ans = -1
32             break
33         tmp *= 2
34     print(ans)
```