```python
import math, re, numpy, pandas

# 1.1
s1 = lambda a, b: a + b
# 2.1
s2 = lambda y: math.ceil(y / 100)
# 3.1
s3 = lambda s: s[::-1] == s
# 4.2
s4 = lambda y: max([a * b for a, b in zip(y[:-1], y[1:])])
# 5.2
s5 = lambda n: (n - 1) ** 2 + n ** 2
# 6.2
s6 = lambda s: max(s) - min(s) - len(set(s)) + 1
# 7.2*
s7 = lambda s: 3 > sum((i >= j) + (i >= k) for i, j, k in zip(s, s[1:], s[2:] + [10 ** 6]))
# 8.2
s8 = lambda x: sum( x[0] + list(map(sum, [[x[i][j] for j in range(len(x[i])) if x[i - 1][j] != 0] for i in range(1, len(x))])))
# 9.3
s9 = lambda y: [i for i in y if len(i) == len(max(y, key=len))]
# 10.3
s10 = lambda s1, s2: sum([min(s1.count(i), s2.count(i)) for i in set(s1)])
# 11.3
s11 = lambda n: sum(list(map(int, str(n)[:int(len(str(n)) // 2)]))) == sum(list(map(int, str(n)[int(len(str(n)) // 2):])))
# 12.3

# 13.3*
s13 = lambda s: eval('"' + s.replace('(', '"+("').replace(')', '")[::-1]+"') + '"')
# 14.4
s14 = lambda a: [sum(a[::2]), sum(a[1::2])]
# 15.4*
s15 = lambda p: ["*" * (len(p[0]) + 2)] + ["*" + i + "*" for i in p] + ["*" * (len(p[0]) + 2)]
# 16.4*
s16 = lambda A, B: sorted(A) == sorted(B) and sum([a != b for a, b in zip(A, B)]) <= 2
# 17.4

# 18.4

# 19.5

# 20.5*
s20 = lambda a: max([abs(a[i]-a[i+1]) for i in range(len(a)-1)])
# 21.5
s21 = lambda s: len(s.split('.')) == 4 and all([n.isdigit() and 0 <= int(n) < 256 for n in s.split('.')])
# 22.5*
s22 = lambda ia: min([i for i in range(2, max(ia)+2) if all([j%i!=0 for j in ia])])
# 23.5*
s23 = lambda image: [[int(sum(sum(x[i:i + 3]) for x in image[j:j + 3]) / 9) for i in range(len(image[j]) - 2)] for j in
                     range(len(image) - 2)]
# 24

# 25

# 26
```

```python
# 27

# 28

# 29

# 30

# 31

# 32

# 33

# 34

# 35

# 36

# 37

# 38.9*
s38 = lambda upSpeed, downSpeed, desiredHeight: 1 if desiredHeight <= upSpeed else (desiredHeight - upSpeed - 1) // ( upSpeed - downSpeed) + 2
# 39.9*
s39 = lambda v1, w1, v2, w2, maxW: max((w1 + w2 <= maxW) * (v1 + v2), (w1 <= maxW) * v1, (w2 <= maxW) * v2)
# 40.9*
s40 = lambda i: re.findall('^\d*', i)[0]
# 41.9
def s41(m, d = 0):
    while len(str(sum(list(map(int, str(m)))))) > 1: m = str(sum(list(map(int, str(m))))); d =d + 1
    return d
# 42.9*
s42 = lambda b, p: abs(ord(b[0]) - ord(p[0])) == abs(ord(b[1]) - ord(p[1]))
# 43.10

# 44

# 45

# 46

# 47

# 48.11
s48 = lambda s: s.isdigit()
# 49

# 50

# 51.11*
s51 = lambda n: max([int(str(n)[:i] + str(n)[i+1:]) for i in range(len(str(n)))])
# 52.12*
```

```
s52 = lambda text: max(re.split('[^a-zA-Z]', text), key=len)
# 53

# 54

# 55

# 56

# 57

# 58

# 59

# 60

# 61

# 62

# 63
```