Zhang Caiqi 18085481D

1.

a.

First, we assume that Sk is the result at the end of the *k*th iteration.

Sk= x + (x+1) + … + (x+k)

And if Sk-1 is correct (2ky-x), Sk is correct, because of the two statement within the while() loop.

Because of the same reason, we can say that if S1 is correct, S2 is correct and Sk is correct.

Let us see whether S1 and S2 are correct:

In the first loop, nextNumber is (x+1), result is x + (x+1). 🡪S1 is correct.

In the second loop, nextNumber is (x+2), result is x + (x+1) + (x+2)🡪S2 is correct.

So, in the kth loop, nextNumber is (x+k), result is x + (x+1) + … + (x+k).

***(loop invariance)***

We can know that S1 and S2 are correct. Therefore, Sk is correct.

b.

We already known that Sk= x + (x+1) + … + (x+k).

So we can know that Sn= x + (x+1) + … + (x+n). n=y-x

We can know that the final result is correct because Sk is correct.

Thus, the function gives the correct result, which is sum(n) = x + (x+1) + … + y

2.

a.

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 0 | 00 | 01 |
| 1 | 01 | 10 |

b.

PROC1(a, b, c)

Inputs: a, b, c, three binary digits.

Output: de, a two-digit number which is the result of a + b + c. If a + b + c<10, d=0.

uv🡨PROC0(a, b)

u1v1🡨PROC0(c, v)

if u=u1=0,

d🡨0 and e🡨v1

else

d🡨1 and e🡨v1

return de

3.

def fun (n):

result=1

if n<2

return 1

else

for i in range(1, n +1):

result🡨result\*i

return result

a🡨0, e🡨0

if ak do,

e🡨e + 1/fun(a)

a🡨a + 1

return e

Explanation：

First, we should define a new function to give us the correct answer of natural numbers’ factorial.

Next, a is 0 and e is 0. We enter the loop and do the loop until we add 1/fun(k) as the last item.

Then we end the loop and get the value of e for n = 0, 1, 2, …, k, which is an estimate of ***e.***