嘉義市第 38 屆中小學科學展覽會

作品說明書

科　　別：數學科

組　　別：國中組

作品名稱：卯吃寅糧－糧食與最遠路程的關係暨公式探討

關 鍵 詞：　探險　、　糧食與最遠路程　、　公式探討

編　　號：

製作說明：

1.說明書封面僅寫科別、組別、作品名稱及關鍵詞。

2.編號由國立臺灣科學教育館統一編列。

3.封面編排由參展作者自行設計。

摘要

研究內容起於：有一些人要到沙漠中去探險，從原點出發且每人帶固定的糧食數(但每人身上糧食數有所限制)。我們假設每天每人走 1 公里會消耗 1 份糧食。途中可把糧食分給他人，但不可將糧食存放途中。在此設定的條件下，若每個人可隨時返回，且決定返回時身上攜帶的糧食數需足夠安全返回到起點。我們討論在以上的規則中，其中一人最遠可走幾公里？

接著我們以上述為基礎，進階討論在不同的規則中是否可找出通用的解法及公式。

研究一開始，我們利用圖紙動手實作來得出相關數據並希望找出其規律性。但發現在規則越來越複雜以及數據越來越大後，手做變得既麻煩又易出錯。因此我們想到利用一年級獨立研究課程所學的 Python 程式寫出檢驗數據的程式內容，讓我們只要輸入不同的數據即可得知每個人所能行走的最遠距離以及剩餘糧食量。

最後，我們利用實驗數據找出不同規則的通式，並用自己寫出的 Python 程式加以檢驗，得到了〈研究一〉每人食量為 1 時的公式以及〈研究二〉有一半人食量為 2 時的規律。我們嘗試將研究推廣到〈研究三〉有一半人食量為 a (即任意數)，並且已經將程式完整地寫出來，但礙於時間因素未能整理出研究三的結果。

壹、研究動機

某次在網路上算數學題時看見沙漠中探險這個題目，覺得這個研究很有趣、可以討論的空間也很大，便開始去找相關資料與類似的研究。進而發現找到的研究範圍有些侷限，故我們希望將研究範圍擴大，且想試試規則若再做延伸是否有機會可以找出其通式。

貳、研究目的

遊戲規則為：有 n 個人，出發時帶 x 天的糧食(每人每天 一份)，每天可深入 1 公里。途中可把糧食分給他人，但每人身上隨時≤ x 份糧食。又不可將糧食存放途中，且每個人可隨時返回，此時身上的糧食需使此人可安全回到起點，求其中一人最遠可走幾公里？

我們做了以下三個研究：

一、研究每人每天吃一份糧食，求其中一人最遠可走公里數之規律。

二、研究其中一半的人改為每天吃 2 份糧食，求其中一人最遠可走公里數之規律。

三、研究其中一半的人改為每天吃 a 份糧食，求其中一人最遠可走公里數之規律。

參、研究設備及器材

一、電腦、紙、筆

二、Word、Excel、Python(程式語言)

肆、研究過程與方法

一、名詞定義

(一)　n：人數

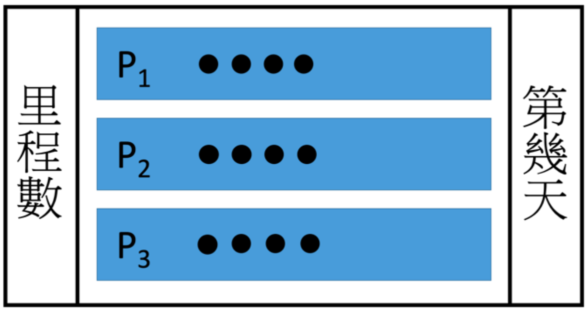
(二)　x：每人可攜帶的最大糧食量

(三)　P_n / A_n / B_n：第 n 個人走的公里數

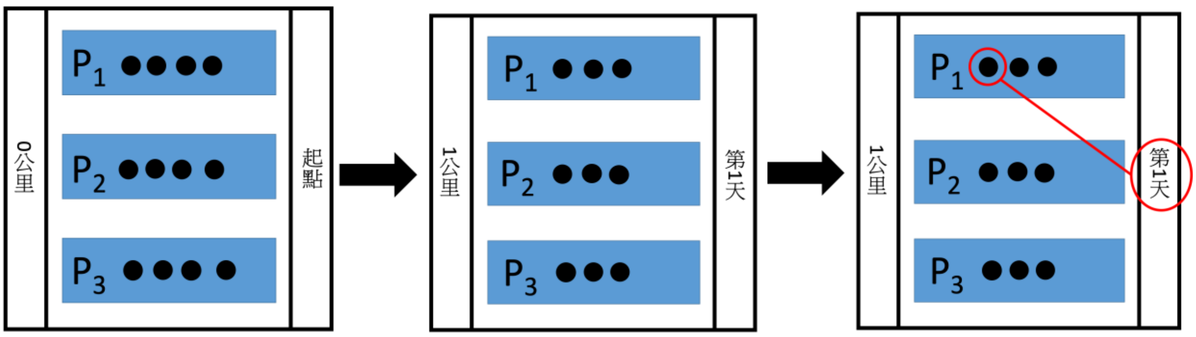
(四)　R_n：P_n 最後剩餘的糧食量

(五)　$\lceil\ \rceil$：天花板函數，即無條件進位

二、利用手作移動找到團體中其中一人可到達之最遠距離：

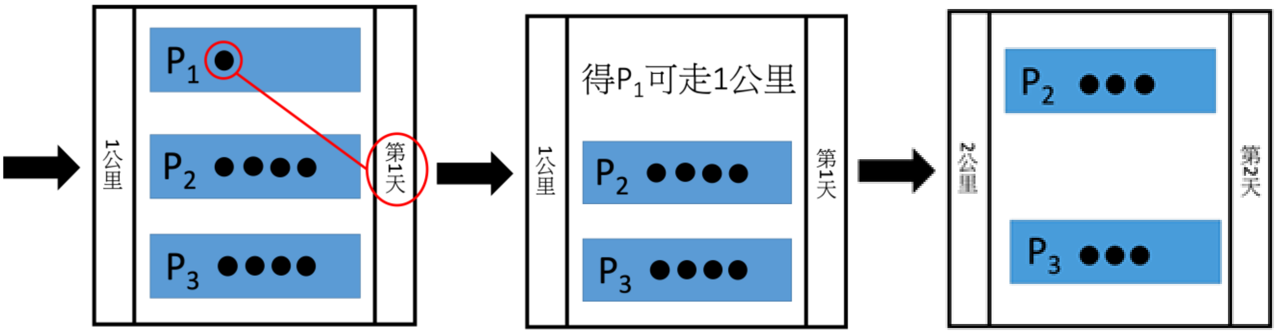


(一) 研究一：以 n=3,x=4 為例



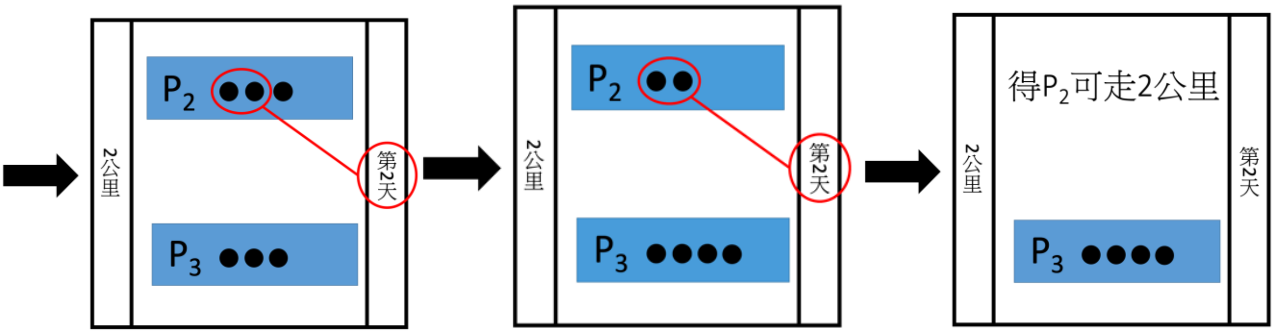
圖形說明：

1. 開始時，每人攜帶 4 份糧食，每往前走 1 公里，每人消耗 1 份糧食(即剩餘 3 份)。
2. P_1 將其餘的糧食儘量分給後面的人，並且 P_1 至少需保留足夠自己回程的糧食量(即 1 份)。



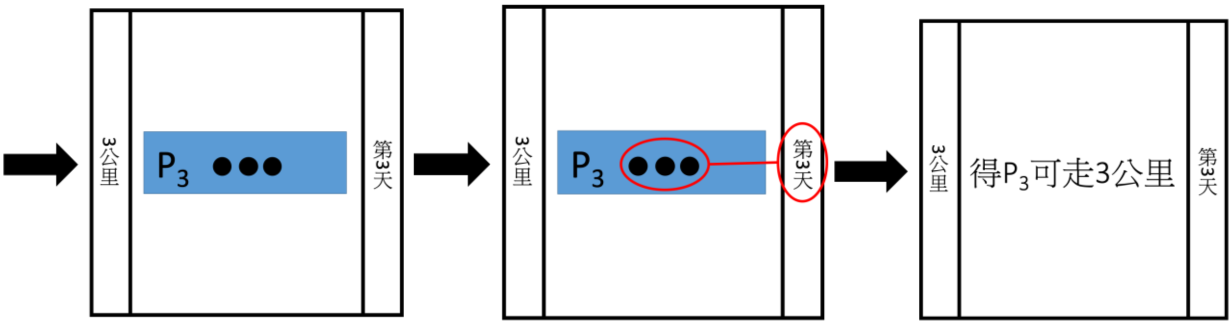
圖形說明：

1. P_1 回起點，$R_1=0$。
2. 其他人繼續往前走 1 公里，每人消耗 1 份糧食 (即剩餘 3 份)。



圖形說明：

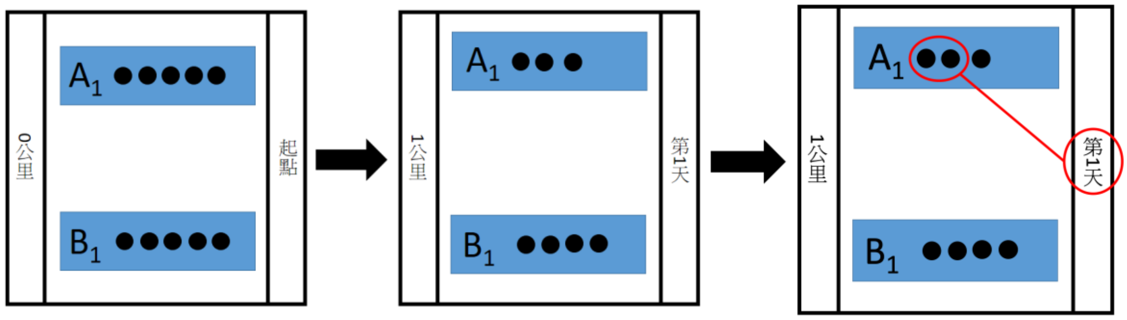
1. P_2 至少需保留足夠自己回程的糧食量(即 2 份)，並將其餘的糧食儘量分給後面的人。
2. P_2 回起點，$R_2=0$。



圖形說明：

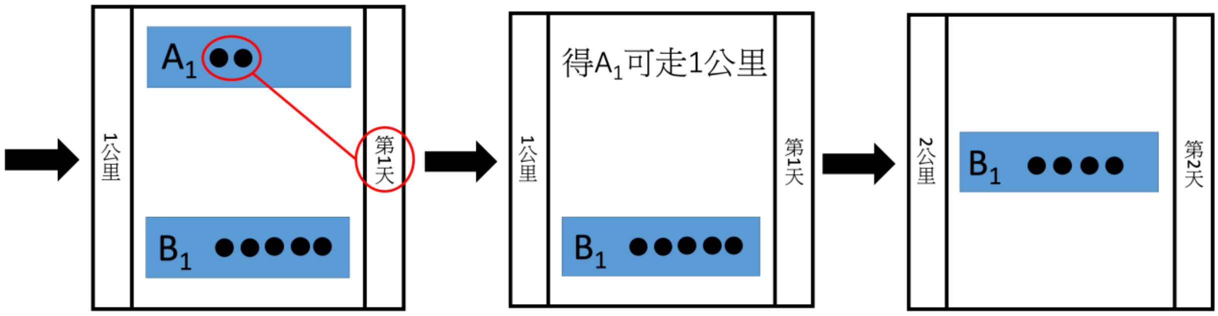
1. 每往前走 1 公里，每人消耗 1 份糧食(即剩餘 3 份)，P_3 的糧食量足夠自己回程 (即 3 份)，但無法再往前走。
2. P_3 回起點，$R_3=0$。P_3 最遠可走 3 公里。

(二) 研究二：以 n=2,x=5 為例



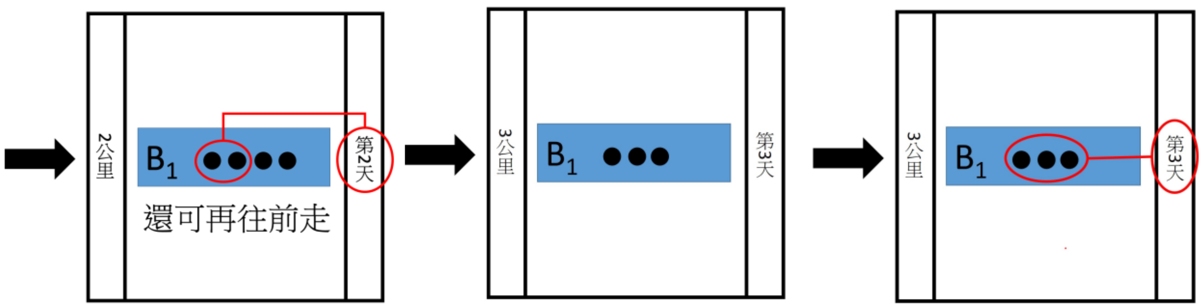
圖形說明：

1. 開始時，每人攜帶 4 份糧食，每往前走 1 公里。
2. A_1 消耗 2 份糧食(即剩餘 3 份)，B_1 消耗 1 份糧食(即剩餘 4 份)。
3. A_1 至少需保留足夠自己回程的糧食量(即 1 份)。



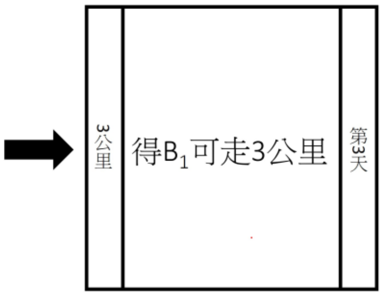
圖形說明：

1. A_1 將其餘的糧食儘量分給後面的人，A_1 回起點，R=0。
2. 繼續往前走 1 公里，B_1 消耗 1 份糧食(即剩餘 4 份)。



圖形說明：

1. B_1 扣除的足夠自己回程糧食量(即 2 份)，還有 2 份，則還可再往前走 1 公里，B_1 消耗 1 份糧食(即剩餘 3 份)。
2. B_1 的糧食量足夠自己回程 (即 3 份)，故無法再往前走，B_1 回起點。



圖形說明：

1. $R_3=0$，故 B_1 最遠可走 3 公里。

三、將實驗過程及數據整理成表格及公式：

(一)、研究一：條件如下

1. 每人每天吃一份糧食，每人皆從原點出發。
2. 每人帶相同的糧食數 x ($x>n$)。
3. 每天每人走 1 公里會消耗 1 份糧食。

(1)　n=2 之實驗數據：(當 $x\leq 2$ 時，無法來回，故不列入表格討論)

x	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
P_1	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
P_2	1	2	2	3	4	4	5	6	6	7	8	8	9	10	10
剩餘(R)	0	0	2	0	0	2	0	0	2	0	0	2	0	0	2

➢ 從表格中數據，我們發現 P_2 所能行走之最大公里數的規律：

a. 當 $x=3k$，則 $P_2=x*2/3$

b. 當 $x=3k+1$，則 $P_2=[(x+2)*2/3]-2$

c. 當 $x=3k+2$，則 $P_2=[(x+1)*2/3]-1$

以上 k 皆為非負整數 ⇒ 以上整理獲得 **公式：$\lceil (x-1)*2/3 \rceil$**

〈Pf〉：

a. $x=3k$，$P_2=3k*2/3=2k$，$\lceil (3k-1)*2/3 \rceil=\lceil 2k-2/3 \rceil=2k$ ⇒$2k=2k$，兩式相等。

b. $x=3k+1$，$P_2=\{[(3k+1)+2]*2/3\}-2=(2k+2)-2=2k$，

$\lceil [(3k+1)-1]*2/3 \rceil=\lceil 2k \rceil=2k$

⇒$2k=2k$，兩式相等。

c. $x=3k+2$，$P_2=\{[(3k+2)+1]*2/3\}-1=(2k+2)-1=2k+1$

$\lceil [(3k+2)-1]*2/3 \rceil=\lceil 2k+2/3 \rceil=2k+1$

⇒$2k+1=2k+1$，兩式相等。　(證明另附手稿：附件一　p.20)

(2) n=3 之實驗數據：(當 x≤3 時，無法來回，故不列入表格討論)

x	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
P_1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
P_2	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8
P_3	2	3	3	4	5	6	6	7	8	9	9	10	11	12	12
剩餘(R)	1	0	3	0	1	0	3	0	1	0	3	0	1	0	3

➢ 從表格中數據，我們發現 P_3 所能行走之最大公里數的規律：

當 x=4k，則 $P_3=x*3/4$

當 x=4k+1，則 $P_3=[(x+3)*3/4]-3$

當 x=4k+2，則 $P_3=[(x+2)*3/4]-2$

當 x=4k+3，則 $P_3=[(x+1)*3/4]-1$

以上 k 皆為非負整數 ⇒ 以上整理獲得 **公式：[(x-1)*3/4]**

(證明另附手稿：附件一 p.20)

(3) n=4 之實驗數據：(當 x≤4 時，無法來回，故不列入表格討論)

x	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
P_1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4
P_2	2	2	2	3	3	4	4	4	5	5	6	6	6	7	7	8	8
P_3	2	3	3	4	5	5	6	6	7	8	8	9	9	10	11	11	12
P_4	3	4	4	5	6	7	8	8	9	10	11	12	12	13	14	15	16
剩餘(R)	0	0	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0

➢ 從表格中數據，我們發現 P_4 所能行走之最大公里數的規律：

當 x=5k，則 $P_4= x*4/5$

當 x=5k+1，則 $P_4=[(x+4)*4/5]-4$

當 x=5k+2，則 $P_4=[(x+3)*4/5]-3$

當 x=5k+3，則 $P_4=[(x+2)*4/5]-2$

當 x=5k+4，則 $P_4=[(x+1)*4/5]-1$

以上 k 皆為非負整數 ⇒ 以上整理獲得 **公式：[(x-1)*4/5]**

(證明另附手稿：附件二 p21)

(4) n=5 之實驗數據：(當 x≤5 時，無法來回，故不列入表格討論)

x	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**

P_1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4
P_2	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6	7
P_3	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10
P_4	3	4	4	5	6	6	7	8	8	9	10	10	11	12	12	13
P_5	4	5	5	6	7	8	9	10	10	11	12	13	14	15	15	16
剩餘(R)	1	0	5	0	1	2	1	0	5	0	1	2	1	0	5	0

- 從表格中數據，我們發現 P_5 所能行走之最大公里數的規律：

當 $x=6k$: $P_5=x*5/6$

當 $x=6k+1$: $P_5=[(x+5)*5/6]-5$

當 $x=6k+2$: $P_5=[(x+4)*5/6]-4$

當 $x=6k+3$: $P_5=[(x+3)*5/6]-3$

當 $x=6k+4$: $P_5=[(x+2)*5/6]-2$

當 $x=6k+5$: $P_5=[(x+1)*5/6]-1$

以上 k 皆為非負整數 ⇒ 以上整理獲得 **公式：[(x-1)*5/6]**

(證明另附手稿：附件三 p.22)

(二)、研究二：將出發的人數對半分成兩組：(目前只討論 n 為偶數時)

一組食量為 2：以「 A_1 , A_2 , … , $A_{n/2}$」表示。

另一組食量為 1：以「B_1 , B_2 , … , $B_{n/2}$」表示。

1. n=2 (當 x≤4 時，無法來回，不列入表格討論)

x	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
A_1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3
B_1	2	3	3	4	4	5	6	6	7	7	8	9	9	10	10	11
剩餘(R)	0	0	2	2	4	0	0	2	2	4	0	0	2	2	4	4

- 從表格中數據，我們發現 B_1 所能行走之最大公里數的規律：

當 $x=5k$: $B_1=x*3/5$

當 $x=5k+1$: $B_1=[(x+4)*3/5]-3$

當 $x=5k+2$: $B_1=[(x+3)*3/5]-2$

當 $x=5k+3$: $B_1=[(x+2)*3/5]-2$

當 $x=5k+4$: $B_1=[(x+1)*3/5]-1$，以上 k 皆為非負整數

2. n=4 (當 x≤5 時，無法來回，不列入表格討論)

x	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A_1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
A_2	1	1	2	2	2	2	2	3	3	3	4	4	4	4	4	5
B_1	2	3	3	4	4	5	5	5	6	7	7	8	8	9	9	9
B_2	3	4	5	6	6	7	8	8	9	10	11	12	12	13	14	14
剩餘(R)	2	2	0	0	4	4	6	2	2	2	0	0	4	4	6	2

➢ 從表格中數據，我們發現 B_2 所能行走之最大公里數的規律：

當 x=8k : B_2=x*6/8

當 x=8k+1 : B_2=[(x+7)*6/8]-6

當 x=8k+2 : B_2=[(x+6)*6/8]-5

當 x=8k+3 : B_2=[(x+5)*6/8]-4

當 x=8k+4 : B_2=[(x+4)*6/8]-4

當 x=8k+5 : B_2=[(x+3)*6/8]-3

當 x=8k+6 : B_2=[(x+2)*6/8]-2

當 x=8k+7 : B_2=[(x+1)*6/8]-1，以上 k 皆為非負整數

3. n=6 (當 x≤4 時，無法來回，不列入表格討論)

x	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A_1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
A_2	1	1	1	1	1	2	2	2	2	2	2	3	3	3
A_3	1	1	1	2	2	2	3	3	3	3	3	4	4	4
B_1	1	2	2	3	3	4	4	5	5	6	6	6	7	7
B_2	2	3	3	4	5	5	6	7	7	8	8	9	10	10
B_3	3	4	4	5	6	7	8	9	9	10	11	12	13	13
剩餘(R)	0	0	6	2	4	2	0	0	6	6	10	0	0	6

➢ 從表格中數據，我們發現 B_3 所能行走之最大公里數的規律：

當 x=11k : B3=x*9/11

當 x=11k+1 : B_3=[(x+10)*9/11]-9

當 x=11k+2 : B_3=[(x+9)*9/11]-8

當 x=11k+3 : B_3=[(x+8)*9/11]-7

當 x=11k+4 : B_3=[(x+7)*9/11]-6

當 x=11k+5 : B_3=[(x+6)*9/11]-5

當 x=11k+6 : B_3=[(x+5)*9/11]-5

當 x=11k+7 : B_3=[(x+4)*9/11]-4

當 x=11k+8 : B_3=[(x+3)*9/11]-3

當 x=11k+9 : B_3=[(x+2)*9/11]-2

當 x=11k+10 : B_3=[(x+1)*3/4]-1，以上 k 皆為非負整數

(三)、 研究三：將出發的人數對半分成兩組：(目前只討論 n 為偶數時)

一組食量為 3：以「 $A_1 , A_2 , \ldots , A_{n/2}$」表示。

另一組食量為 1：以「$B_1 , B_2 , \ldots , B_{n/2}$」表示。

1. n=2 (當 x≤6 時，無法來回，不列入表格討論)

x	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
A_1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4	4
B_1	3	4	4	5	5	6	6	7	8	8	9	9	10	10	11	12	12	13	13	14	14	15	16
剩餘(R)	0	0	2	2	4	4	6	0	0	2	2	4	4	6	0	0	2	2	4	4	6	0	0

➢ 從表格中數據，我們發現 B_1 所能行走之最大公里數的規律：

當 x = 7k+1 : B_1=[(x+6)*4/7]-4

當 x = 7k+2 : B_1=[(x+5)*4/7]-3

當 x = 7k+3 : B_1=[(x+4)*4/7]-3

當 x = 7k+4 : B_1=[(x+3)*4/7]-2

當 x = 7k+5 : B_1=[(x+2)*4/7]-2

當 x = 7k+6 : B_1=[(x+1)*4/7]-1，以上 k 皆為非負整數

2. n=4 (當 x≤3 時，無法來回，不列入表格討論)

x	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
A_1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	3
A_2	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	4	4	4	5
B_1	2	3	3	4	4	5	5	6	6	6	7	7	8	8	9	9	10	10	11	11	11	12	12
B_2	4	5	5	6	7	8	8	9	10	10	11	12	13	13	14	15	16	16	17	18	18	19	20
剩餘(R)	0	0	4	4	0	0	4	4	6	10	10	0	0	4	4	0	0	4	4	6	10	10	0

➢ 從表格中數據，我們發現 B_2 所能行走之最大公里數的規律：

當 x = 11k : B_2=x*8/11

當 x = 11k+1 : B_2=[(x+10)*8/11]-8

當 $x = 11k+2$: $B_2=[(x+9)*8/11]-7$

當 $x = 11k+3$: $B_2=[(x+8)*8/11]-6$

當 $x = 11k+4$: $B_2=[(x+7)*8/11]-6$

當 $x = 11k+5$: $B_2=[(x+6)*8/11]-5$

當 $x = 11k+6$: $B_2=[(x+5)*8/11]-4$

當 $x = 11k+7$: $B_2=[(x+4)*8/11]-3$

當 $x = 11k+8$: $B_2=[(x+3)*8/11]-3$

當 $x = 11k+9$: $B_2=[(x+2)*8/11]-2$

當 $x = 11k+10$: $B_2=[(x+1)*8/11]-1$，以上 k 皆為非負整數

四、利用 Python 程式設計檢驗數據：

(一) 研究一：每人每天吃一份糧食，每人皆從原點出發。每人帶相同的糧食數 x ($x>n$)。我們假設每天每人走 1 公里會消耗 1 份糧食。

程式內容如下：

```
print("n 人出發探險，每人帶滿食物上限 x，最遠可以走多遠？")
n=int(input("人數(n)="))
x=int(input("可攜帶糧食上限(x)="))
min=int(1)    # 標號最小的人的編號
R=0       # 餘糧
day=0   # 第 day 天
k=n

p=[x]*(n+1)                          # 宣告 p 陣列，可容 n 變數，都先填入 x
                                     # p[a] 表示第 a 人所擁有的糧食數

while(1):              # 開始迴圈 day 表天數
    while(1):
        if(p[min]==day+1):
            print("第",min,"人走到",day,"公里")
            min+=1
            R+=1
        elif(p[min]==day):
            print("第",min,"人走到",day,"公里")
            min+=1
        if(min==n):
            R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
            print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
            print("糧食剩餘",int(R),"天份")
            import sys                    #結束程式
            sys.exit(0)
        else:
            break

    day+=1

    for i in range(min,n+1):             # 每天的開始每人耗用一份糧食
        p[i]+=-1
    k=n

    while(1):

        if(p[k]==x):                     #檢查可收糧的人的編號
            k-=1
            if(k==min):
```

```
                break
            elif(k==min+1):
                if(p[min]==day):
                    print("第",min,"人走到",day,"公里")
                    min+=1
                    break

        while(1):
            while(1):
                if(p[min]==day):                    #檢查 p[min]是否能給
                    print("第",min,"人走到",day,"公里")
                    min+=1
                    if(n==min):
                        R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                        print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                        print("糧食剩餘",int(R),"天份")
                        import sys                  #結束程式
                        sys.exit(0)
                else:
                    break
            if(min==n):
                R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                print("糧食剩餘",int(R),"天份")
                import sys
                sys.exit(0)
            else:
                break
        if(k==min):                                 #檢查
            break

        p[k]+=1
        p[min]-=1
```

(二) 研究二：其中一半的人改為每人每天吃 2 份糧食，其餘規定不變。

程式內容如下：

```
print("n 人出發探險，每人帶滿食物上限 x，且有一半的人一天必須吃 2 份食物，最遠可以走多遠？")
print("P.S.人數必須為偶數")
n=int((input("人數(n)=")))
x=int(input("可攜帶糧食上限(x)="))
if(int(n/2)!=n/2):
    print("人數必須為偶數哦")
```

```
        import sys                          #結束程式
        sys.exit(0)
min=int(1)   # 標號最小的人的編號
R=0      # 餘糧
day=0   # 第 day 天
p=[x]*(n+1)
while(1):
        while(1):
                if(min>=(n/2)+1):
                        if(p[min]==day+1):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                                R+=1
                        elif(p[min]==day):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                        if(min==n):
                                R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                                print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                                print("糧食剩餘",int(R),"天份")
                                import sys                  #結束程式
                                sys.exit(0)
                        else:
                                break
                elif(min<=(n/2)):
                        if(p[min]==2*day):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                        elif(p[min]==2*day+1):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                                R+=1
                        elif(p[min]==2*day+2):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                                R+=2
                        elif(p[min]==2*day+3):
                                print("第",min,"人走到",day,"公里")
                                min+=1
                                R+=3
                        else:
                                break

        day+=1

        for i in range(min,n+1):            # 每天的開始每人耗用一份糧食
```

```
        if(i>=(n/2)+1):
            p[i]+=-1
            k=n
        elif(i<=(n/2)):
            p[i]+=-2
            k=n
    while(1):
        if(p[k]==x):                              #檢查可收糧的人的編號
            k-=1
            if(k==min):
                break
            elif(k==min+1):
                if(min>=(n/2)+1):
                    if(p[min]==day):
                        print("第",min,"人走到",day,"公里")
                        min+=1
                        break
                if(min<=(n/2)):
                    if(p[min]==2*day):
                        print("第",min,"人走到",day,"公里")
                        min+=1
                        break

        while(1):
            while(1):
                if(min>=(n/2)+1):
                    if(p[min]==day):                              #檢查 p[min]是否能給
                        print("第",min,"人走到",day,"公里")
                        min+=1
                        if(n==min):
                            R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                            print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                            print("糧食剩餘",int(R),"天份")
                            import sys                              #結束程式
                            sys.exit(0)
                    else:
                        break
                elif(min<=(n/2)):
                    if(p[min]==2*day):
                        print("第",min,"人走到",day,"公里")
                        min+=1
                    else:
                        break
            if(min==n):
                R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
```

```
                print("糧食剩餘",int(R),"天份")
                import sys
                sys.exit(0)
            else:
                break
        if(k==min):                          #檢查
            break

        p[k]+=1
        p[min]-=1
```

(三) 研究三：其中一半的人改為每天吃 a 份糧食，其餘規定不變。

程式內容如下：

```
print("n 人出發探險，每人帶滿食物上限 x，且有一半的人一天必須吃 a 份食物，最遠可以走多遠？")
print("P.S.人數必須為偶數")
n=int((input("人數(n)=")))
x=int(input("可攜帶糧食上限(x)="))
a=int(input("一半人每天所需的糧食份量(a)="))
if(int(n/2)!=n/2):
    print("人數必須為偶數哦")
    import sys                    #結束程式
    sys.exit(0)
min=int(1)   # 標號最小的人的編號
R=0      # 餘糧
day=0   # 第 day 天
p=[x]*(n+1)
while(1):
    while(1):
        if(min>=(n/2)+1):
            if(p[min]==day+1):
                print("第",min,"人走到",day,"公里")
                min+=1
                R+=1
            elif(p[min]==day):
                print("第",min,"人走到",day,"公里")
                min+=1
            if(min==n):
                R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                print("糧食剩餘",int(R),"天份")
                import sys                    #結束程式
```

```
                sys.exit(0)
            else:
                break
        elif(min<=(n/2)):
            g=min
            for i in range(a):
                if(p[min]==a*day+i):
                    print("第",min,"人走到",day,"公里")
                    min+=1
                    R+=i
                    break
            if(min==g):
                break

day+=1

for i in range(min,n+1):                    #  每天的開始每人耗用一份糧食
    if(i>=(n/2)+1):
        p[i]+=-1
        k=n
    elif(i<=(n/2)):
        p[i]+=-a
        k=n
while(1):
    if(p[k]==x):                            #檢查可收糧的人的編號
        k-=1
        if(k==min):
            break
        elif(k==min+1):
            if(min>=(n/2)+1):
                if(p[min]==day):
                    print("第",min,"人走到",day,"公里")
                    min+=1
                    break
            if(min<=(n/2)):
                if(p[min]==a*day):
                    print("第",min,"人走到",day,"公里")
                    min+=1
                    break

    while(1):
        while(1):
            if(min>=(n/2)+1):
                if(p[min]==day):                #檢查 p[min]是否能給
                    print("第",min,"人走到",day,"公里")
                    min+=1
```

```
                    if(n==min):
                        R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
                        print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
                        print("糧食剩餘",int(R),"天份")
                        import sys                      #結束程式
                        sys.exit(0)
                else:
                    break
            elif(min<=(n/2)):
                if(p[min]==a*day):
                    print("第",min,"人走到",day,"公里")
                    min+=1
                else:
                    break
        if(min==n):
            R += 2*(0.5*(p[min]-day) - int(0.5*(p[min]-day)))
            print("第",min,"人走到",day+int(0.5*(p[min]-day)),"公里")
            print("糧食剩餘",int(R),"天份")
            import sys
            sys.exit(0)
        else:
            break
    if(k==min):                          #檢查
        break

    p[k]+=1
    p[min]-=1
```

伍、研究結果

一、〈研究一〉當每人每天吃一份糧食時，我們整理數據得到下列通式：

$x=(n+1)k$: $P_n=x*n/(n+1)$

$x=(n+1)k+1$: $P_n=[(x+n)*n/(n+1)]-n$

$x=(n+1)k+2$: $P_n=[(x+(n-1))*n/(n+1)]-(n-1)$

$x=(n+1)k+3$: $P_n=[(x+(n-2))*n/(n+1)]-(n-2)$

(以此類推)……

$x=(n+1)k+n$: $P_n=[(x+1)*n/(n+1)]-1$

註: $k\in N$ or $k=0$

⇒ **公式：$\lceil (x-1)*n/(n+1) \rceil$** (證明另附手稿：附件四 p.23)

二、〈研究二〉當有一半人數每天吃兩份糧食時，我們整理數據得到下列通式：

(一) n=4m 時

當 $x=(3n/2+2)k$: $P_n=x*[3n/2/(3n/2+2)]$

當 $x=(3n/2+2)k+1$: $P_n=[x+(3n/2+1)]*[3n/2/(3n/2+2)]-(3n/2)$

當 $x=(3n/2+2)k+2$: $P_n=[x+(3n/2+0)]*[3n/2/(3n/2+2)]-(3n/2-1)$

當 $x=(3n/2+2)k+3$: $P_n=[x+(3n/2-1)]*[3n/2/(3n/2+2)]-(3n/2-2)$

(以此類推，直到 $x=(3n/2+2)k+[(3n/2+2)/2]$ 時，減數重複上一個減數，之後繼續類推)

當 $x=(3n/2+2)k+(3n/2+1)$: $P_n=[x+1]*[3n/2/(3n/2+2)]-1$

(二) n=4m+2 時

當 $x=(3n/2+2)k$: $P_n=x*[3n/2/(3n/2+2)]$

當 $x=(3n/2+2)k+1$: $P_n=[x+(3n/2+1)]*[3n/2/(3n/2+2)]-(3n/2)$

當 $x=(3n/2+2)k+2$: $P_n=[x+(3n/2+0)]*[3n/2/(3n/2+2)]-(3n/2-1)$

當 $x=(3n/2+2)k+3$: $P_n=[x+(3n/2-1)]*[3n/2/(3n/2+2)]-(3n/2-2)$

(以此類推，直到 $x=(3n/2+2)k+[(3n/2+2)/2]+1$ 時，減數重複上一個減數，之後繼續類推)

當 $x=(3n/2+2)k+(3n/2+1)$: $P_n=[x+1]*[3n/2/(3n/2+2)]-1$

註: $k\in N$ or $k=0$，$m\in N$ or $m=0$

陸、未來展望

我們在討論過程中發現，根據狀況的不同、改變不同的實驗變數，會有不同的規律出現，進而可以推出通式。在實驗過程中發現數據的變化實在有趣，因此為了尋找和確認較大變數的規律是否成立，我們想到用程式驗算當作輔助證明。但也因為如此，我們花了許多時間驗證，目前〈研究三〉我們還來不及找出適合的通式，故沒有將研究三的結果列入此次報告中。

如果時間足夠，很希望有機會可以討論不同的人數比例，不同的食量，進而推導出更多的結果。我們相信這在現實生活中，應該可以被加以運用的，例如探險前糧食的預估……

且上學期末我們在做期末報告時，彰師大數學系教授對我們的建議與指導，讓我們也思考或許可以用數學歸納法證明我們用數據規律導出的通式，好讓導出的公式更加確定而不只是推測。

以上便是我們接下來想繼續往下做的研究方向。

柒、參考資料及其他

一、陳鈺婷、葉于甄 43 屆國小組科展參展作品–沙漠任務

二、類似題 2008 年青少年數學國際城市邀請賽–個人賽試題

三、類似題 YLL 討論網 國小挑戰題 6 題試題

四、108 年 12 月到彰師大數學系報告，教授提議可試著以天花板符號表示通式。

五、109 年 03 月參加中正大學辦理「雲嘉地區中小學科學教育揚升計畫」，獲得指導。

$n=2$。

x	P_1	P_2	R
2	1	1	0
3	1	2	0
4	1	2	2
5	2	3	0
6	2	4	0
7	2	4	2
8	3	5	0
9	3	6	0
10	3	6	2
11	4	7	0
12	4	8	0
13	4	8	2
14	5	9	0
15	5	10	0
16	5	10	2

證明。

$x=3K \quad P_2=\underline{2K}=\lceil (3K-1)\times\frac{2}{3}\rceil$

$=\lceil 2K-\frac{2}{3}\rceil=\underline{2K}$

$x=3K+1 \quad P_2=\underline{2K}=\lceil (3K)\times\frac{2}{3}\rceil=\underline{2K}$

$x=3K+2 \quad P_2=\underline{2K+1}=\lceil (3K+1)\times\frac{2}{3}\rceil$

$=\lceil 2K+\frac{2}{3}\rceil=\underline{2K+1}$

探討。

$x=3K \quad P_2=\frac{2}{3}x$

$x=3K+1 \quad P_2=\frac{2}{3}(x+2)-2$

$x=3K+2 \quad P_2=\frac{2}{3}(x+1)-1$

推導。

$x=3K \quad P_2=\frac{2}{3}\cdot 3K=2K$ 不進位

$x=3K+1 \quad P_2=\frac{2}{3}(3K+3)-2=2K$ 不進

$x=3K+2 \quad P_2=\frac{2}{3}(3K+3)-1=2K+1$ 進位

推測。

K係數 $3\to 2 \Rightarrow \times\frac{2}{3}$

前2個不進位

$x=3K+1 \quad P_2=2K$

$\Rightarrow P_2=\lceil (x-1)\times\frac{2}{3}\rceil$

$n=3$

x	P_1	P_2	P_3	R
3	1	1	2	1
4	1	2	3	0
5	1	2	3	3
6	2	3	4	0
7	2	3	5	1
8	2	4	6	0
9	2	4	6	3
10	3	5	7	0
11	3	5	8	1
12	3	6	9	0
13	3	6	9	3
14	4	7	10	0
15	4	7	11	1
16	4	8	12	0
17	4	8	12	3

探討

$x=4K \quad P_3=\frac{3}{4}x$

$x=4K+1 \quad P_3=\frac{3}{4}(x+3)-3$

$x=4K+2 \quad P_3=\frac{3}{4}(x+2)-2$

$x=4K+3 \quad P_3=\frac{3}{4}(x+1)-1$

推導

$x=4K \quad P_3=\frac{3}{4}\cdot 4K=3K$

$x=4K+1 \quad P_3=\frac{3}{4}(4K+4)-3=3K$

$x=4K+2 \quad P_3=\frac{3}{4}(4K+4)-2=3K+1$

$x=4K+3 \quad P_3=\frac{3}{4}(4K+4)-1=3K+2$

證明

$x=4K \quad P_3=\underline{3K}=\lceil (4K-1)\times\frac{3}{4}\rceil$

$=\lceil 3K-\frac{3}{4}\rceil=\underline{3K}$ #

$x=4K+1 \quad P_3=\underline{3K}=\lceil (4K)\times\frac{3}{4}\rceil=\underline{3K}$ #

$x=4K+2 \quad P_3=\underline{3K+1}=\lceil (4K+1)\times\frac{3}{4}\rceil$

$=\lceil 3K+\frac{3}{4}\rceil=\underline{3K+1}$ #

$x=4K+3 \quad P_3=\underline{3K+2}=\lceil (4K+2)\times\frac{3}{4}\rceil$

$=\lceil 3K+\frac{6}{4}\rceil=\underline{3K+2}$ #

推測

比照 $n=2$ 公式

推測可能為 $\underline{P_3=\lceil (x-1)\times\frac{3}{4}\rceil}$

$n=4$

x	P_1	P_2	P_3	P_4	R
4	1	2	2	3	0
5	1	2	3	4	0
6	1	2	3	4	4
7	2	3	4	5	0
8	2	3	5	6	0
9	2	4	5	7	0
10	2	4	6	8	0
11	2	4	6	8	4
12	3	5	7	9	0
13	3	5	8	10	0
14	3	6	8	11	0
15	3	6	9	12	0
16	3	6	9	12	4
17	4	7	10	13	0
18	4	7	11	14	0
19	4	8	11	15	0
20	4	8	12	16	0

探討

$x=5K \quad P_4=\frac{4}{5}x$

$x=5K+1 \quad P_4=\frac{4}{5}(x+4)-4$

$x=5K+2 \quad P_4=\frac{4}{5}(x+3)-3$

$x=5K+3 \quad P_4=\frac{4}{5}(x+2)-2$

$x=5K+4 \quad P_4=\frac{4}{5}(x+1)-1$

推導

$x=5K \quad P_4=\frac{4}{5}\cdot 5K=4K$

$x=5K+1 \quad P_4=\frac{4}{5}(5K+5)-4=4K$

$x=5K+2 \quad P_4=\frac{4}{5}(5K+5)-3=4K+1$

$x=5K+3 \quad P_4=\frac{4}{5}(5K+5)-2=4K+2$

$x=5K+4 \quad P_4=\frac{4}{5}(5K+5)-1=4K+3$

推測

比照 $n=2$、$n=3$ 公式

推測可能為 $P_4=\lceil (x-1)\times\frac{4}{5}\rceil$

證明

$x=5K \quad P_4=\underline{4K}=\lceil (5K-1)\times\frac{4}{5}\rceil$
$=\lceil 4K-\frac{4}{5}\rceil=\underline{4K}$

$x=5K+1 \quad P_4=\underline{4K}=\lceil (5K)\times\frac{4}{5}\rceil=\underline{4K}$

$x=5K+2 \quad P_4=\underline{4K+1}=\lceil (5K+1)\times\frac{4}{5}\rceil$
$=\lceil 4K+\frac{4}{5}\rceil=\underline{4K+1}$

$x=5K+3 \quad P_4=\underline{4K+2}=\lceil (5K+2)\times\frac{4}{5}\rceil$
$=\lceil 4K+\frac{8}{5}\rceil=\underline{4K+2}$

$x=5K+4 \quad P_4=\underline{4K+3}=\lceil (5K+3)\times\frac{4}{5}\rceil$
$=\lceil 4K+\frac{12}{5}\rceil=\underline{4K+3}$

研究結果：

$x=(n+1)K \quad P_n=\frac{n}{n+1}x=nK$

$x=(n+1)K+1 \quad P_n=\frac{n}{n+1}(x+n)-n=nK$

$x=(n+1)K+2 \quad P_n=\frac{n}{n+1}[x+(n-1)]-(n-1)=nK+1$

$x=(n+1)K+3 \quad P_n=\frac{n}{n+1}[x+(n-2)]-(n-2)=nK+2$

$x=(n+1)K+4 \quad P_n=\frac{n}{n+1}[x+(n-3)]-(n-3)=nK+3$

$\vdots$

$x=(n+1)K+n \quad P_n=\frac{n}{n+1}[x+1]-1=nK+(n-1)$

推測公式為 $\underline{P_n=\left\lceil (x-1)\times\frac{n}{n+1}\right\rceil}$

證明：

$x=(n+1)K \quad P_n=\underline{nK}=\left\lceil [(n+1)K-1]\times\frac{n}{n+1}\right\rceil=\left\lceil nK-\frac{n}{n+1}\right\rceil=\underline{nK}\ \#$

$x=(n+1)K+1 \quad P_n=\underline{nK}=\left\lceil [(n+1)K]\times\frac{n}{n+1}\right\rceil=\underline{nK}\ \#$

$x=(n+1)K+2 \quad P_n=\underline{nK+1}=\left\lceil [(n+1)K+1]\times\frac{n}{n+1}\right\rceil=\left\lceil nK+\frac{n}{n+1}\right\rceil=\underline{nK+1}\ \#$

$x=(n+1)K+3 \quad P_n=\underline{nK+2}=\left\lceil [(n+1)K+2]\times\frac{n}{n+1}\right\rceil=\left\lceil nK+\frac{2n}{n+1}\right\rceil=\left\lceil nK+2+\frac{-2}{n+1}\right\rceil=\underline{nK+2}\ \#$

$\vdots$

$x=(n+1)K+m \quad P_n=\underline{nK+(m-1)}=\left\lceil [(n+1)K+(m-1)]\times\frac{n}{n+1}\right\rceil=\left\lceil nK+\frac{(m-1)n}{n+1}\right\rceil=\left\lceil nK+(m-1)+\underline{\frac{-(m-1)}{n+1}}\right\rceil=\underline{nK+(m-1)}\ \#$

$(1\le m\le n$ 且 $m\in N)$

$1\le m\le n$

$\Rightarrow -(n-1)\le -(m-1)\le 0$

$\Rightarrow -1<\frac{-(m-1)}{n+1}\le 0$

$\vdots$

$x=(n+1)K+n \quad P_n=\underline{nK+(n-1)}=\left\lceil [(n+1)K+n-1]\times\frac{n}{n+1}\right\rceil=\left\lceil nK+\frac{(n-1)n}{n+1}\right\rceil=\left\lceil nK+n-1+\frac{-(n-1)}{n+1}\right\rceil=\underline{nK+(n-1)}\ \#$