\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW2

Author: F74071140 謝育萱 <[shirleysensible@gmail.com](mailto:shirleysensible@gmail.com)>

Class: 甲班

Description:

這次的完成時間果然比較久，剛開始不太相信才學完簡單的迴圈就可以寫出看似複雜的小遊戲。起初連讓使用者連續輸入數字都覺得奇怪，因為第一次把scanf放在迴圈用，中途在計算X的地方卡好久，還跟朋友做了一些討論，發現每個人的方法和想法都不太一樣，但邏輯是大同小異的，對我來說最有效率的作業方法是與他人交換意見，還有分段跑程式比較容易找到錯誤。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

int main(int argc ,char\* argv[])

{

int N=atoi(argv[1]) ,P=atoi(argv[2]);

printf("You set N=%d ,P=%d\n",N ,P); /\*print:N= P= \*/

int p[P],g[P],i=0,j=0;

do

{

printf("ans:");

for(i=0;i<P;i++)

{

scanf("%d",&p[i]); /\*輸入答案\*/

}

for(i=0;i<P;i++) /\*檢查輸入的數字是否符合條件\*/

{

if(p[i]<N+1&&p[i]>0)

j=1; /\*符合條件離開do-while loop\*/

else

{

printf("The answer is illegal,please reset the answer.\n");

j=0; /\*不符合條件則重新輸入答案\*/

break;

}

}

}while(j==0);

while(j==1)

{

int H=0 ,X=0;

printf("guess:");

for(i=0;i<P;i++)

{

scanf("%d",&g[i]); /\*輸入猜測數字\*/

}

for(i=0;i<P;i++)

{

if(g[i]==p[i]) /\*數有幾個數字對且位置對的猜測\*/

H++;

}

int count\_ans=0,count\_guess=0,k=0;

for(k=0;k<N+1;k++)

{

int sum;

for(i=0;i<P;i++) /\*數有幾個數字對的猜測(不論位置對與否)\*/

{

if(p[i]==k) /\*統計答案有幾個1,2,...,N\*/

count\_ans++;

if(g[i]==k) /\*統計猜測有幾個1,2,...,N\*/

count\_guess++;

}

sum=count\_ans<=count\_guess?count\_ans:count\_guess; /\*找兩者間較小的統計結果\*/

count\_ans=0; /\*歸零繼續下一次統計\*/

count\_guess=0;

X=X+sum;

}

X=X-H; /\*數有幾個數字對但位置不對的猜測\*/

printf("%dH %dX\n",H ,X); /\*print: ?H ?X\*/

if(H==P&&X==0)

j=3; /\*跳出while loop\*/

}

return 0;

Compilation:

gcc hw0\_3.c -o hw3

Execution:

./ hw0\_3 5 5

Output:

You set N=5 ,P=5

ans:0 2 3 7 3

The answer is illegal,please reset the answer.

ans:2 3 4 5 5

guess:1 2 3 4 5

1H 3X

guess:4 5 4 5 4

2H 1X

guess:2 3 4 5 5

5H 0X

F74071140@c-2018-2:~/hw0>