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

Report: HW4\_2

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

Class: 甲班

Description:

Homeworks#2

Function\_1

測試輸入答案是否符合條件。

當輸入的數值全落在1~N之間，結束for迴圈後j=1，由於j=1不符合while迴圈的條件，因此跳出迴圈；若輸入值其中有一個沒有在1~N之間，則以j=0跳出for迴圈後重跑while迴圈。

Function\_2

數有幾個位置對且數字對的猜測（計算有幾個H）。

分別放入兩個值ans（ans為p陣列中的數）與guess（guess為g陣列中的數），若兩值相等，則H的值多1。

Function\_3

統計答案／猜測值各有幾個1~N的數。

若輸入值ans（為p陣列或g陣列）等於數字k（介於1~N之間），則把C加上1以累計有幾個數為數字對（暫且不論其位置），並回傳C值。

Function\_4

找兩者間較小的統計結果。

若輸入值a較小，則回傳a值；反之則回傳a值。

Homeworks#3

Function\_1

把字元轉換成整數陣列。

做if的是非判斷，如果輸入字元為｀1＇，則b儲存1；反之則存0。

Function\_2

將num的各位元修改成我們想要的樣子。

（利用bitwise|及<<運算法，與1做|運算為1；與0做|運算則保持原來值）a從num(為0)的最左位元比對,若a為1其位元由0轉為1。

Function\_3

呈現num的位元。

（利用bitwise中&<<運算法，與1做&運算保持原樣；與0做&則為0）1從num最左的bit比對,若同為1,n的該bit存1,反之存0。

Function\_4

呈現num的2進位儲存方式。

Function\_5

將字串轉存為陣列。

做if的是非判斷，若輸入字元為｀1＇則b存1；反之則存0。

Function\_6

計算整數型態x的位元值。

（利用bitwise|及<<運算法，與1做|運算為1；與0做|運算則保持原來值）x從最左bit開始比對,若b為1則其位元轉為1。

Function\_7

呈現num的位元值。

（利用bitwise中&<<運算法，與1做&運算保持原樣；與0做&則為0）1以long long型別存取,從最左開始比對,若同為1則該bit存1,反之存0將該bit值右移至底,使p值為該bit值。

Function\_8

計算輸入值a的位元值。

（利用bitwise中&<<運算法，與1做&運算保持原樣；與0做&則為0）用long long來表示其位元組成的\*y，將其每個位元依序跟1做&運算將該bit值右移至底,使其值變為該bit值。

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

#include<stdio.h>

int function\_1(int a,int n)

{

int l;

if(a<n+1&&a>0)

{ l=1; } /\*符合條件離開do-while loop\*/

else

{

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

l=0; /\*不符合條件則重新輸入答案\*/

}

return l;

}

int function\_2(int H,int guess,int ans) /\*數有幾個位置對且數字對的猜測\*/

{

if(guess==ans)

H++;

return H;

}

int function\_3(int k,int ans,int c) /\*統計答案/猜測值各有幾個1~N的數\*/

{

if(ans==k)

c++;

return c;

}

int function\_4(int a,int b) /\*找兩者間較小的統計結果\*/

{

if(a<b)

return a;

else

return b;

}

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;

while(j==0)

{

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\*/

{

j=function\_1(p[i],N); /\*呼叫函式\*/

if(j==0)

break;

}

}

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++)

{

H=function\_2(H,g[i],p[i]); /\*呼叫函式\*/

}

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

for(k=1;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)\*/

{

count\_ans=function\_3(k,p[i],count\_ans); /\*呼叫函式\*/

count\_guess=function\_3(k,g[i],count\_guess); /\*呼叫函式\*/

}

sum=function\_4(count\_ans,count\_guess); /\*呼叫函式\*/

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 hw4\_2.c -o hw4\_2

Execution:

F74071140@c-2018-2:~/hw4> ./hw4\_2 11 99

You set N=11 ,P=99

ans:6 8 8 4 3 3 1 8 9 10 4 10 10 9 5 3 4 2 10 8 6 9 4 5 4 2 8 2 10 10 9 8 9 2 2 1 9 7 8 5 9 1 5 2 3 9 4 5 10 1 4 3 4 3 10 2 6 8 5 10 10 5 6 10 10 11 4 8 7 2 4 6 2 7 5 6 4 8 6 2 5 4 9 5 10 7 1 10 1 10 4 10 10 1 7 5 10 4 11

guess:6 9 3 2 3 8 7 8 1 2 2 5 2 1 1 1 6 1 8 5 2 1 1 7 2 7 3 1 9 6 3 8 6 8 7 7 2 9 4 7 2 8 4 9 4 8 6 7 2 9 9 9 3 5 7 9 8 4 7 7 1 4 9 5 6 4 7 8 6 8 5 4 1 1 3 8 2 9 8 7 3 3 6 2 2 6 9 4 7 4 6 2 8 8 2 7 5 6 4

5H 65X

guess:5 5 1 3 3 2 6 1 7 2 6 4 7 2 6 7 2 6 4 3 3 3 3 1 1 2 6 7 4 3 6 3 5 5 2 3 2 7 3 2 4 3 4 7 4 2 3 7 4 5 3 2 6 4 2 1 2 1 3 7 2 2 6 1 4 5 2 4 5 6 7 5 7 7 4 5 4 2 5 1 2 4 7 1 6 2 6 1 1 4 4 6 3 5 3 1 1 2 4

10H 50X

guess:1 2 1 2 3 3 3 1 1 2 3 1 3 3 1 3 1 1 1 1 3 1 2 2 1 1 2 1 3 3 2 2 3 3 3 1 2 1 3 1 2 2 3 1 1 2 3 2 2 2 1 2 1 1 1 1 2 3 3 2 1 3 1 2 1 3 1 1 2 2 1 1 1 1 1 3 1 2 3 2 1 3 2 3 1 1 2 2 3 2 2 3 1 1 2 2 1 2 2

7H 16X

guess:6 8 8 4 3 3 1 8 9 10 4 10 10 9 5 3 4 2 10 8 6 9 4 5 4 2 8 2 10 10 9 8 9 2 2 1 9 7 8 5 9 1 5 2 3 9 4 5 10 1 4 3 4 3 10 2 6 8 5 10 10 5 6 10 10 11 4 8 7 2 4 6 2 7 5 6 4 8 6 2 5 4 9 5 10 7 1 10 1 10 4 10 10 1 7 5 10 4 11

99H 0X

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

Report: HW4\_3

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

Class: 甲班

Description:

（參照HW4\_2處）

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

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int function\_1(char a) //將字元轉換成整數0,1儲存

{

int b;

if(a=='1')

b=1;

else

b=0;

return b;

}

int function\_2(int num,int a,int i) //a從num(為0)的最左位元比對,若a為1其位元由0轉為1

{

if(a==1)

num|=a<<(31-i);

return num;

}

int function\_3(int num,int i) //1從num最左的bit比對,若同為1,n的該bit存1,反之存0

{

int n;

n=num&(1<<i);

n>>=i;

return n;

}

int function\_4(int num,int i) //1從num最左的bit比對,若同為1,n的該bit存1,反之存0

{

int n=num&(1<<i);

n>>=i;

return n;

}

long long function\_5(char a) //將字串轉存為陣列

{

long long b;

if(a=='1')

b=1;

else

b=0;

return b;

}

long long function\_6(long long b,long long x,int i) //x從最左bit開始比對,若b[i]為1則其位元轉為1

{

if(b==1)

x=x|b<<(63-i);

return x;

}

long long function\_7(long long num,int i) //1以long long型別存取,從最左開始比對,若同為1則該bit存1,反之存0將該bit值右移至底,使p值為該bit值

{

long long p;

p=num&((long long)1<<i);

p>>=i;

return p;

}

long long function\_8(long long a,int i) //用long long來表示其位元組成的\*y，將其每個位元依序跟1做&運算將該bit值右移至底,使其值變為該bit值

{

long long bit;

bit=a&((long long)1<<i);

bit>>=i;

return bit;

}

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

{

int i=0, n=0;

switch(atoi(argv[1]))

{

case 1:

{

float f=0.0; //將float值初始化

int \*k=(int\*)&f; //f轉為整數指標,並把其位址存於k

int a[32];

for(i=0;i<32;i++)

{

a[i]=function\_1(argv[2][i]); //呼叫函式

}

for(i=0;i<32;i++)

{

n=function\_2(n,a[i],i); //呼叫函式

(\*k)=function\_2((\*k),a[i],i);

}

printf("integer:%d\n",n); //print n

printf("float:%f\n",f); //print f

break;

}

case 2:

{

int num=atoi(argv[2]); //用atoi將字串轉為整數

printf("32-bit:");

for(i=31;i>-1;i--)

{

n=function\_3(num,i); //呼叫函式

printf("%d",abs(n));

}

printf("\n");

break;

}

case 3:

{

float a=atof(argv[2]); //用atof將字串轉為float

int \*x=(int\*)&a; //將a轉整數指標並存取其位址

printf("32-bit:");

for(i=31;i>=0;i--)

{

n=function\_4((\*x),i); //呼叫函式

printf("%d",abs(n)); //print n的絕對值

}

printf("\n");

break;

}

case 4:

{

long long b[64];//為避免溢位使用long long宣告，底下有long long皆同此理

for(i=0;i<64;i++)

b[i]=function\_5(argv[2][i]); //呼叫函式

long long x=0;

for(i=63;i>=0;i--)

{

x=function\_6((long long)b[i],x,i); //呼叫函式

}

double \*d=(double\*)&x; //將long long x轉double

printf("integer:%lld\n",x);

printf("double:%f\n",\*d);

break;

}

case 5:

{

long long num=atoll(argv[2]); //用atol將字串argv[2]轉為long int

printf("64-bit:");

for(i=63;i>=0;i--)

{

long long p=function\_7(num,i); //呼叫函式

printf("%d",abs(p)); //print p的絕對值

}

printf("\n");

break;

}

case 6:

{

double d=strtod(argv[2],NULL); //將字串轉換為double

long long \*y=(long long\*)&d; //運用指標將double位元以long long表示

printf("64-bit:");

for(i=63;i>=0;i--)

{

long long bit=function\_8((\*y),i); //呼叫函式

printf("%lld",abs(bit));

}

printf("\n");

}

}

return 0;

}

Compilation:

gcc hw4\_3c -o hw4\_3

Execution:

F74071140@c-2018-2:~/hw4> ./hw4\_3 1 01001000011100110110010100011111

integer:1215522079

float:249236.484375

F74071140@c-2018-2:~/hw4> ./hw4\_3 2 -3811651

32-bit:11111111110001011101011010111101

F74071140@c-2018-2:~/hw4> ./hw4\_3 3 90568.491822

32-bit:01000111101100001110010000111111

F74071140@c-2018-2:~/hw4> ./hw4\_3 4 110000100001110101000010001000111111110000011111100101101111011

integer:-4459335333705705738

double:-31415926535.897423

F74071140@c-2018-2:~/hw4> ./hw4\_3 5 8223372036854775808

64-bit:0111001000011111010010010100110001011000100111000000000000000000

F74071140@c-2018-2:~/hw4> ./hw4\_3 6 -4459335333705705605.123456

64-bit:1100001111001110111100010101111011101110000000011111000000110100

F74071140@c-2018-2:~/hw4>