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

Report: HW2

Author: E64061151 林友鈞 <[E64061151@mail.ncku.edu.tw](mailto:wang@xiaoming.tw)>

Class: 資訊系二乙

Description:

我從這份作業學到如何使用scnaf讀含有空白的字串，以及在使用scanf時須清理buffer，否則會影響到下一次的scanf。另外我從這份作業也學到如何切割字串。

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

Code:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<time.h>

// ever number enters from user is an object.

// Attribute:

struct number{

int num;

int match; // record the number has ever match or not

};

struct userdata{

struct number data[100];

int amount;

};

int SetData(struct userdata \*data,const char \*choice){

// The function of SetData is to get the input from user.

//

// Parameter:

// struct userdata \*data:

// This parameter is a pointer, which can be "guess" or "ans".

//

// const char \*choice:

// This parameter is a string, the api user must specify which kind of input they want.

// For example: pass "ans:" or pass "guess:" in to the function

//

// Description:

// The design method I used is get the string type of numbers by scanf.

// I split the string based on blank(space) and I use atoi to conver the string to integer.

// This method I can easily catch the input from user no matter how many numbers the user wants to enter.

char str\_data[200]; // access the data input from user

char \*pch;

int i = 0,scf;

printf("%s",choice);

scf = scanf("%[^\n]",str\_data); // get input data, use %[^\n] to get string with "space"

while ((getchar()) != '\n'); // clear the buffer to avoid influencing next input

if(scf == 0 || scf == EOF){ // if input data is illegal to scanf

return 0;

}

else{

// if input data is legal, the program will split the string based on space

pch = strtok(str\_data," ");

while(pch!=NULL){

data->data[i].num = atoi(pch);

data->data[i].match = 0;

pch = strtok(NULL," ");

i++;

}

data->amount = i;

// the amount of number of input from user is i which is iterator and also

// means that how many numbers are split out.

}

// if pass all steps ,function will return 1 finally.

return 1;

}

int Check(struct userdata \*data,int P,int N){

// function Check is to check if the input from user is legal to this game or not.

//

// parameter:

// struct userdata \*data:

// This parameter is a pointer which can be "ans" or "guess".

// int P:

// P is input from command line

// int N:

// N is input from command line

int i;

// check the amount of the numbers is legal or not. If the amount of the input is not equal to N, the function will return 0

if(data->amount != P){

// printf("amount doesn't match\n");

return 0;

}

// check if the input is within the range of N.

// if find a number beyond the range function Check will return 0.

for(i = 0; i < data->amount;i++){

if(data->data[i].num > N || data->data[i].num < 1){

// printf("%d beyond the range",data->data[i].num);

return 0;

}

}

// if input pass all check, return 1.

return 1;

}

int HX(struct userdata ans,struct userdata \*guess,int P){

// This function will determine the H and X.

//

// Parameter:

// struct userdata ans:

// The user must pass ans into the function.

// struct userdata \*guess:

// This parameter is a pointer, it point to the original guess which is declared in main function.

// int P:

// P is input from command line.

int i = 0,j = 0;// iterator

int h = 0,x = 0;

// The first step is to find the H. The number and position of number in guess and ans should be the same.

// if match,guess->data[i].match and ans.data[j].match will be assigned 1

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

for(j = 0;j < P; j++){

if(ans.data[j].match == 0){ // this program means that ans.data[j] has not matched

if(guess->data[i].num == ans.data[j].num && i == j){

h++;

guess->data[i].match = ans.data[j].match = 1;

}

}

}

}

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

if(guess->data[i].match == 0){

for(j = 0; j < P; j++){

if(ans.data[j].match == 0 && ans.data[j].num == guess->data[i].num){

x++;

ans.data[j].match = 1;

}

}

}

}

printf("%dH%dX\n",h,x);

return h;

}

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

{

int N = atoi(argv[1]);

int P = atoi(argv[2]);

printf("You set N=%d, P=%d\n",N,P);

struct userdata ans,guess;

// Play!

while(!SetData(&ans,"ans:")||!Check(&ans,P,N))

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

do{

while(!SetData(&guess,"guess:")||!Check(&guess,P,N));

}while(P != HX(ans,&guess,P));

return 0;

}

Compilation:

gcc -o hw2 hw2.c

Execution:

./hw2 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

1H3X

guess:4 5 4 5 4

2H1X

guess:1 4 3 5 5

2H2X

guess:2 4 3 5 5

3H2X

guess:2 3 4 5 5

5H0X