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

Report: HW4

Author: F74071027 學貝萁 <youqing1211@gmail.com>

Class: 甲班

Description:

Homework #2

int checkq(int check,int ans[p]): 檢查輸入答案是否合乎規定，如果不符合則回傳0。

int calculation(int gue[p],int ans[p],int ansnum[n+1],int guenum[n+1]): 計算總共幾H和幾X，猜對回傳1使程式結束，猜錯則回傳0，程式繼續。

Homework #3

void b2if(int Q[32]): 模式一，將32-bit轉為int，後交由printif輸出與轉成float輸出。

void i2b(int Q): 模式二，將int轉為32-bit，後交由print32b輸出。

void f2b(float Q): 模式三，將float轉為int再轉成32-bit，後交由print32b輸出。

void b2llid(long long int Q[64]):模式四，將64-bit轉成long long int，後交由printllid輸出與轉成double輸出。

void lli2b(long long int Q): 模式五，將long long int轉為64-bit，後交由print64b輸出。

void d2b(double Q): 模式六，將double轉成long long int再轉為64-bit，後交由print64b輸出。

void printif(int A): 輸出模式一的int，再轉成float來輸出。

void printllid(long long int A): 輸出模式四的long long int，再轉成double來輸出。

void print32b(int n,int A[32]): 輸出模式二和模式三的32-bit。

void print64b(long long int n,long long int A[64]): 輸出模式五和模式六的64-bit。

----------------------

　　這次作業主要是要練習使用函式分割程式，讓程式的可讀性提高。但在將沒有函式的程式碼使用函式分割時要記得注意哪些變數需要轉成全域變數，再加上這次要分割出12個函式（只有我覺得12個太多了嗎），所以在傳值上也要很注意，慢慢檢查有沒有哪裡有資料型態不同的問題。

　　另外，函式通常需要擺在main的前面才有辦法編譯，放在main後面則會編譯失敗（但是這個server跑得過去而且只有float在傳值時有問題，我很訝異也不知道為何）。如果想要放在main後面就要事先宣告，像我的程式碼。

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

**HW4\_2:**

Code:

#include<stdio.h>

#include<stdlib.h>

int n,p,i; //n:play with //p:position

int checkq(int check,int ans[p]);

int calculation(int gue[p],int ans[p],int ansnum[n+1],int guenum[n+1]);

///////////////////////////////////////////////

///////////////////////////////////////////////

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

n = atoi(argv[1]);

p = atoi(argv[2]);

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

int ans[p];

while(1>0){

printf("ans: ");

for(i=0;i<p;i++)scanf("%d",&ans[i]);

int check = 1;

check=checkq(check,ans);

if(check==1)break;

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

}

//======================================

while(1>0){

int gue[p], ansnum[n+1], guenum[n+1];

for(i=0;i<n+1;i++){

ansnum[i]=0;

guenum[i]=0;

}

printf("guess: ");

for(i=0;i<p;i++)scanf("%d",&gue[i]);

//==========calculation==========

int judge = calculation(gue,ans,ansnum,guenum);

if(judge==1)break;

}

return 0;

}

////////////////////////////////////////

////////////////////////////////////////

int checkq(int check,int ans[p]){ //check if ans is illegal

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

if(ans[i]>n||ans[i]<1){

check = 0;

break;

}

}

return check;

}

int calculation(int gue[p],int ans[p],int ansnum[n+1],int guenum[n+1]){

int h=0,x=0;

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

if(gue[i]==ans[i])h++;

else{

ansnum[ans[i]]++;

guenum[gue[i]]++;

}

}

if(h==p){

printf("%dH 0X\n",h);

return 1;

}

else{

for(i=1;i<n+1;i++){

if(guenum[i]<ansnum[i])x=x+guenum[i];

else x=x+ansnum[i];

}

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

return 0;

}

}

Compilation:

gcc hw4\_2.c -o hw4\_2

Execution:

./hw4\_2 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

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

**HW4\_3:**

Code:

#include<stdio.h>

#include<stdlib.h>

int M,i,j;

//Mode

void b2if(int Q[32]); //32-bit > int & float

void i2b(int Q); //int > 32-bit

void f2b(float Q); //float > 32-bit

void b2llid(long long int Q[64]); //64-bit > int & double

void lli2b(long long int Q); //int > 64-bit

void d2b(double Q); //double > 64-bit

//Print

void printif(int A); //print int float

void printllid(long long int A); //print int double

void print32b(int n,int A[32]); //print 32-bit

void print64b(long long int n,long long int A[64]); //print 64-bit

///////////////////////////////////////////////

///////////////////////////////////////////////

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

M=atoi(argv[1]);

if(M==1){ //32-bit > int & float

int Q[32];

for(i=0;i<32;i++)Q[i]=argv[2][i]-48;

b2if(Q);

}

else if(M==2){ //int > 32-bit

int Q=atoi(argv[2]);

i2b(Q);

}

else if(M==3){ //float > 32-bit

float Q=atof(argv[2]);

f2b(Q);

}

else if(M==4){ //64-bit > int & double

long long int Q[64];

for(i=0;i<64;i++)Q[i]=argv[2][i]-48;

b2llid(Q);

}

else if(M==5){ //int > 64-bit

long long int Q=atoll(argv[2]);

lli2b(Q);

}

else if(M==6){ //double > 64-bit

double Q=atof(argv[2]);

d2b(Q);

}

return 0;

}

///////////////////////////////////////////////

///////////////////////////////////////////////

void b2if(int Q[32]){ //32-bit > int & float

int A=0,n=1;//negative

if(Q[0]==0)n=0; //positive

for(i=n;i<32;i++)if(Q[i]==1){

for(j=0;j<31-i;j++)Q[i]=Q[i]\*2;

A=A+Q[i];

}

if(n==1)A=A-2147483648;

printif(A);

}

void i2b(int Q){ //int > 32-bit

int A[32]={0},arrow=31,n=0;//positive

if(Q<0){

Q=Q+2147483648;

n=1;//negative

}

while(Q!=0){

A[arrow]=Q%2;

Q=Q/2;

arrow--;

}

print32b(n,A);

}

void f2b(float Q){ //float > 32-bit

int \*pQ;

pQ=&Q;

int Q1=\*pQ;

//======

int A[32]={0},arrow=31,n=0;//positive

if(Q1<0){

Q1=Q1+2147483648;

n=1;//negative

}

while(Q1!=0){

A[arrow]=Q1%2;

Q1=Q1/2;

arrow--;

}

print32b(n,A);

}

void b2llid(long long int Q[64]){ //64-bit > int & double

long long int A=0,n=1;//negative

if(Q[0]==0)n=0; //positive

for(i=n;i<64;i++)if(Q[i]==1){

for(j=0;j<63-i;j++)Q[i]=Q[i]\*2;

A=A+Q[i];

}

if(n==1)A=A-9223372036854775808;

printllid(A);

}

void lli2b(long long int Q){ //int > 64-bit

long long int A[64]={0},arrow=63,n=0;//positive

if(Q<0){

Q=Q+9223372036854775808;

n=1;//negative

}

while(Q!=0){

A[arrow]=Q%2;

Q=Q/2;

arrow--;

}

print64b(n,A);

}

void d2b(double Q){ //double > 64-bit

long long int \*pQ;

pQ=&Q;

long long int Q1=\*pQ;

//======

long long int A[64]={0},arrow=63,n=0;//positive

if(Q1<0){

Q1=Q1+9223372036854775808;

n=1;//negative

}

while(Q1!=0){

A[arrow]=Q1%2;

Q1=Q1/2;

arrow--;

}

print64b(n,A);

}

/////////////////////////////////////////

/////////////////////////////////////////

void printif(int A){ //print int float

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

float \*pA;

pA=&A;

printf("float: %f\n",\*pA);

}

void printllid(long long int A){ //print int double

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

double \*pA;

pA=&A;

printf("double: %lf\n",\*pA);

}

void print32b(int n,int A[32]){ //print 32-bit

if(n==0)printf("32-bit: ");

else printf("32-bit: 1");

for(i=n;i<32;i++)printf("%d",A[i]);

printf("\n");

}

void print64b(long long int n,long long int A[64]){ //print 64-bit

if(n==0)printf("64-bit: ");

else printf("64-bit: 1");

for(i=n;i<64;i++)printf("%lld",A[i]);

printf("\n");

}

Compilation:

gcc hw4\_3.c -o hw4\_3

Execution:

./hw4\_3 1 11100000000000000000000000000000

integer: -536870912

float: -36893488147419103232.000000

./hw4\_3 2 111

32-bit: 00000000000000000000000001101111

./hw4\_3 3 111

32-bit: 01000010110111100000000000000000

./hw4\_3 4 0100000000101001100000000000000000000000000000000000000000000000

integer: 4623367229960880128

double: 12.750000

./hw4\_3 5 4623367229960880128

64-bit: 0100000000101001100000000000000000000000000000000000000000000000

./hw4\_3 6 12.75

64-bit: 0100000000101001100000000000000000000000000000000000000000000000