**reality.c**

#include <stdio.h>

#include <stdlib.h>

void reality\_1\_example\_1();

void reality\_1\_example\_2();

void reality\_3();

double fun();

typedef struct {

int a[2];

double d;

} struct\_t;

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

int n; /\* used to loop for reality\_3() \*/

reality\_1\_example\_1();

reality\_1\_example\_2();

if (argc==1) n = 5; /\* default 5 iter \*/

else n = atoi(argv[1]);

reality\_3(n);

return 0;

}

void reality\_1\_example\_1(){

float f=50000, total\_f;

int i=40000, total\_i;

/\* fill here \*/

total\_f = f \* f;

printf("reality\_1\_example\_1\_float: f=%f\n", f);

if(total\_f >= 0) {

printf("f\*f=%f >= 0\n\n", total\_f);

} else { printf("f\*f=%f < 0\n\n", f\*f);}

total\_i = i \* i;

printf("reality\_1\_example\_1\_int: f=%d\n", i);

if(total\_i >= 0) {

printf("i\*i=%d >= 0\n\n", total\_i);

} else { printf("i\*i=%d < 0\n\n", i\*i); }

i=50000;

/\* fill here \*/

total\_i = i \* i;

printf("reality\_1\_example\_1\_int: f=%d\n", i);

if(total\_i >= 0) {

printf("i\*i=%d >= 0\n\n", total\_i);

} else { printf("i\*i=%d < 0\n\n", i\*i); }

}

void reality\_1\_example\_2(){

float fx=1e20,fy=-1e20,fz=3.14;

unsigned int uix=12,uiy=34,uiz=56;

int six=12,siy=34,siz=56;

/\* fill here \*/

printf("reality\_1\_example\_2\_unsigned: uix=%d uiy=%d uiz=%d\n", uix, uiy, uiz);

unsigned int sum1\_u = (uix+uiy)+uiz, sum2\_u = uix+(uiy+uiz);

if(sum1\_u == sum2\_u) {

printf("((uix+uiy)+uiz)=%d == (uix+(uiy+uiz))=%d\n\n", sum1\_u, sum2\_u);

} else { printf("((uix+uiy)+uiz)=%d != (uix+(uiy+uiz))=%d\n\n", sum1\_u, sum2\_u); }

printf("reality\_1\_example\_2\_signed: six=%d siy=%d siz=%d\n", six, siy, siz);

int sum1\_i = (six+siy)+siz, sum2\_i = six+(siy+siz);

if(sum1\_i == sum2\_i) {

printf("((six+siy)+siz)=%d == (six+(siy+siz))=%d\n\n", sum1\_i, sum2\_i);

} else { printf("((six+siy)+siz)=%d != (six+(siy+siz))=%d\n\n", sum1\_i, sum2\_i); }

printf("reality\_1\_example\_2\_float: fx=%1.6e fy=%1.6e fz=%1.6f\n", fx, fy, fz);

float sum1\_f = (fx+fy)+fz, sum2\_f = fx + (fy+fz);

if(sum1\_f == sum2\_f) {

printf("((fx+fy)+fz)=%1.6e == (fx+(fy+fz))=%1.6e\n\n", sum1\_f, sum2\_f);

} else { printf("((fx+fy)+fz)=%1.6e != (fx+(fy+fz))=%1.6e\n\n", sum1\_f, sum2\_f); }

}

void reality\_3(int n) {

double d;

int i;

/\* fill here \*/

printf("reality\_3: n=%d iterations\n", n);

for(i = 0; i < n; i++) d = fun(i);

}

double fun(int i){

volatile struct\_t s;

s.d = 3.14;

s.a[i] = 1073741824;

printf("fun(%d) -> %f\n",i,s.d);

return s.d;

}

/\* End of file \*/

A screenshot of a computer screen

AI-generated content may be incorrect.

**reality4.c**

#include <stdio.h>

#include <stdlib.h>

#include <sys/time.h>

#include <stdint.h>

#define NNN 2048

void copyij();

void copyji();

void init\_mat();

int64\_t time\_diff();

int src[NNN][NNN], dst[NNN][NNN];

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

int i,j,n;

long int del\_sec,del\_msec;

struct timeval tv\_s,tv\_e;

init\_mat();

gettimeofday(&tv\_s, NULL);

copyij();

gettimeofday(&tv\_e, NULL);

/\* fill here to compute elapsed time \*/

del\_sec = tv\_e.tv\_sec - tv\_s.tv\_sec, del\_msec = tv\_e.tv\_usec - tv\_s.tv\_usec;

printf("copyji(): dim=%d: elapsed=%ld.%03ld secs\n",NNN,del\_sec,del\_msec/1000);

init\_mat();

gettimeofday(&tv\_s, NULL);

copyji();

gettimeofday(&tv\_e, NULL);

/\* fill here to compute elapsed time \*/

del\_sec = tv\_e.tv\_sec - tv\_s.tv\_sec, del\_msec = tv\_e.tv\_usec - tv\_s.tv\_usec;

printf("copyji(): dim=%d: elapsed=%ld.%03ld secs\n",NNN,del\_sec,del\_msec/1000);

return 0;

}

void copyij(){

int i,j;

/\* fill here \*/

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

for(j = 0; j < NNN; j++)

dst[i][j] = src[i][j];

}

void copyji(){

int i,j;

/\* fill here \*/

for (j = 0; j < NNN; j++)

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

dst[i][j] = src[i][j];

}

void init\_mat(){

int i,j;

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

for (j=0;j<NNN;j++) src[i][j] = dst[i][j] = 1;

}

A black screen with white text

AI-generated content may be incorrect.