```
#include "stdio.h"
#include "stdlib.h"

void reality1();
void reality2();
void reality3(int number);
double fun(int number);

typedef struct {
   int a[2];
   double d;
} struct_t;


int main(int argc, char *argv[]) {
   int number;
   reality1();
   reality2();

   // Get user input for n
   if (argc == 1)
         number = 5;
   else
         number = atoi(argv[1]);
   reality3(number);

   return 0;
}

void reality1() {
   float fnumber = 50000;
   int inumber   = 40000;

   float fsquared;
   int   isquared;

   // Reality 1 test 1 with floating number
   printf("reality_1_example_1_float: f = %f\n", fnumber);
   printf("f*f=");

   fsquared = fnumber * fnumber;
   if (fsquared >= 0)
         printf("%f which is greater than 0\n\n", fsquared);
   else
```

```
        printf("%f which is less than 0\n\n", fsquared);

    // Reality 1 test 1 with int test 1
    printf("reality_1_example_1_int: i = %d\n", inumber);
    printf("i*i=");

    isquared = inumber * inumber;
    if (isquared >= 0)
        printf("%d which is greater than 0\n\n", isquared);
    else
        printf("%d which is less than 0\n\n", isquared);

    // Reality 1 test 1 with int test 2
    inumber = 50000;
    printf("reality_1_example_1_int: i = %d\n", inumber);
    printf("i*i=");

    isquared = inumber * inumber;
    if (isquared >= 0)
        printf("%d which is greater than 0\n\n", isquared);
    else
        printf("%d which is less than 0\n\n", isquared);
}

void reality2() {
    float fx = 1e20, fy = -1e20, fz = 3.14;
    unsigned int uix = 12, uiy = 34, uiz = 56;
    int          six = 12, siy = 34, siz = 56;

    // Reality 1 test 2 with unisnged int
    printf("reality_1_example_2_unsigned: uix = %d, uiy = %d, uiz = %d\n", uix, uiy, uiz);
    unsigned int eq1 = ((uix + uiy) + uiz), eq2 = (uix + (uiy + uiz));
    printf("((uix + uiy) + uiz)=%d == ((uix + (uiy + uiz))=%d\n\n", eq1, eq2);

    // Reality 1 test 2 with signed int
    printf("reality_1_example_2_signed: six = %d, siy = %d, siz = %d\n", six, siy, siz);
    int eq3 = ((six + siy) + siz), eq4 = (six + (siy + siz));
    printf("((six + siy) + siz)=%d == ((six + (siy + siz))=%d\n\n", eq3, eq4);

    // Reality 1 test 2 with float
    printf("reality_1_example_2_float: fx = %f, fy = %f, fz = %f\n", fx, fy, fz);
    float eq5 = ((fx + fy) + fz), eq6 = (fx + (fy + fz));
    printf("((fx + fy) + fz)=%f != ((fx + (fy + fz))=%f\n\n", eq5, eq6);
```

```
}

void reality3(int number) {
    double d;
    int i;

    printf("reality_3: n = %d iterations\n", number);
    for (int i = 0; i <= number; i++) {
            fun(i);
    }
}

double fun(int number) {
    volatile struct_t s;

    s.d = 3.14;
    s.a[number] = 1073741824;
    printf("fun(%d) -> %f\n", number, s.d);

    return s.d;
}
```

```
shane in ~/Documents/School/Junior/CS350/hw1-arcaro-shane$ ./reality 10
reality_1_example_1_float: f = 50000.000000
f*f=2500000000.000000 which is greater than 0

reality_1_example_1_int: i = 40000
i*i=1600000000 which is greater than 0

reality_1_example_1_int: i = 50000
i*i=-1794967296 which is less than 0

reality_1_example_2_unsigned: uix = 12, uiy = 34, uiz = 56
((uix + uiy) + uiz)=102 == ((uix + (uiy + uiz))=102

reality_1_example_2_signed: six = 12, siy = 34, siz = 56
((six + siy) + siz)=102 == ((six + (siy + siz))=102

reality_1_example_2_float: fx = 100000002004087734272.000000, fy = -100000002004087734272.000000, fz = 3.140
000
((fx + fy) + fz)=3.140000 != ((fx + (fy + fz))=0.000000

reality_3: n = 10 iterations
fun(0) -> 3.140000
fun(1) -> 3.140000
fun(2) -> 3.140000
fun(3) -> 2.000001
fun(4) -> 3.140000
fun(5) -> 3.140000
fun(6) -> 3.140000
*** stack smashing detected ***: terminated
Aborted (core dumped)
shane in ~/Documents/School/Junior/CS350/hw1-arcaro-shane$
```

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <stdint.h>

#define NNN 2048

void copyij();
void copyji();
void init_mat();
double time_diff();

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

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

  double elapsed_time;
  struct timeval tv_s,tv_e;

  init_mat();
  gettimeofday(&tv_s, NULL);
  copyij();
  gettimeofday(&tv_e, NULL);

  /* fill here to compute elapsed time */
  elapsed_time = time_diff(tv_s, tv_e);
  printf("copyji(): dim=%d: elapsed=%03f secs\n",NNN, elapsed_time);


  init_mat();
  gettimeofday(&tv_s, NULL);
  copyji();
  gettimeofday(&tv_e, NULL);
  elapsed_time = time_diff(tv_s, tv_e);
  /* fill here to compute elapsed time */

  printf("copyji(): dim=%d: elapsed=%03f secs\n",NNN, elapsed_time);

  return 0;
}

void copyij(){
  int i,j;
```

```
  for (int i = 0; i < NNN; i++) {
     for (int j = 0; j < NNN; j++) {
           dst[i][j] = src[i][j];
     }
  }

}

void copyji(){
  int i,j;

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

}

double time_diff(struct timeval start, struct timeval end) {
    double elapsed_time = end.tv_sec - start.tv_sec + (end.tv_usec - start.tv_usec) * 1e-6;
    return elapsed_time;
}
```

```
shane in ~/Documents/School/Junior/CS350/hw1-arcaro-shane$ ./reality4
copyji(): dim=2048: elapsed=0.011569 secs
copyji(): dim=2048: elapsed=0.095457 secs
shane in ~/Documents/School/Junior/CS350/hw1-arcaro-shane$
```