**Anosh Abraham**

OUTPUT
reality.png

```
[appa@fedora hw1-abraham-anosh]$ ./reality 5
reality_1_example_1_float: f=50000.000000
f*f=2500000000.000000 >= 0

reality_1_example_1_int: i=40000
i*i=1600000000 >= 0

reality_1_example_1_int: i=50000
i*i=-1794967296 < 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=1.000000e+20 fy=-1.000000e+20 fz=3.140000

((fx+fy)+fz)=3.140000e+00 != (fx+(fy+fz))=0.000000e+00

reality_3: n=5 iterations
fun(0) -> 3.140000
fun(1) -> 3.140000
fun(2) -> 3.140000
fun(3) -> 2.000001
fun(4) -> 3.140000
Segmentation fault (core dumped)
```

reality4.png

```
[appa@fedora hw1-abraham-anosh]$ ./reality4
copyji(): dim=2048: elapsed=0.094 secs
copyji(): dim=2048: elapsed=0.295 secs
[appa@fedora hw1-abraham-anosh]$
```

reality.c

```
[appa@fedora hw1-abraham-anosh]$ cat 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;
  int i=40000;

  /* fill here */
  printf("reality_1_example_1_float: f=%f\n",f);
  printf("f*f=%f >= 0\n\n",f*f);
  printf("reality_1_example_1_int: i=%d\n",i);
  printf("i*i=%d >= 0\n\n",i*i);

  i=50000;

  /* fill here */
  printf("reality_1_example_1_int: i=%d\n",i);
  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;
```

```
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);
  printf("((uix+uiy)+uiz)=%d == (uix+(uiy+uiz))=%d\n\n", ((uix+uiy)+uiz), (uix+(uiy+uiz)));

  printf("reality_1_example_2_signed six=%d siy=%d siz=%d\n", six, siy, siz);
  printf("((six+siy)+siz)=%d == (six+(siy+siz))=%d\n\n", ((six+siy)+siz), (six+(siy+siz)));

  printf("reality_1_example_2_float fx=%e fy=%e fz=%f\n\n", fx, fy, fz);
  printf("((fx+fy)+fz)=%e != (fx+(fy+fz))=%e\n\n", ((fx+fy)+fz), (fx+(fy+fz)));
}

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++) 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 */
[appa@fedora hw1-abraham-anosh]$
```

reality4.c

```
[appa@fedora hw1-abraham-anosh]$ cat 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_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(){
```

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

  /* fill here */
  for (i = 0; i < 2048; i++)
    for (j = 0; j < 2048; j++)
      dst[i][j] = src[i][j];
}

void copyji(){
  int i,j;

  /* fill here */
  for (j = 0; j < 2048; j++)
    for (i = 0; i < 2048; 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;
}
[appa@fedora hw1-abraham-anosh]$
```

Makefile

```
[appa@fedora hw1-abraham-anosh]$ cat Makefile
OBJS = reality.o reality4.o
CC = gcc
PROGS = reality reality4

all:    $(PROGS)

reality:    reality.o
        $(CC) reality.c -o reality
reality4:    reality4.o
        $(CC) reality4.c -o reality4
clean:
        rm *.o *~ reality reality4

[appa@fedora hw1-abraham-anosh]$
```