

[Pre-Requisite Workshop-1]

Student Id : NP03A190051

Student Name : Rishab Sharma

Section : L5CG7

Lecturer : Jnaneshwor Bohara

Module leader : Hiran Patel

Submitted on : 18th September, 2021

1. bitwise01

#include <stdio.h>

void toBinary(unsigned char c, unsigned char \*result) {

unsigned char i;

unsigned char mask = 128;

for(i=0;i<8;i++){

if(mask & c) {

result[i]='1';

} else {

result[i]='0';

}

mask = mask >> 1;

}

result[8]=0;

}

int main() {

char result1[9];

char result2[9];

char result3[9];

char result4[9];

char result5[9];

char result6[9];

char result7[9];

char result8[9];

unsigned char n;

unsigned char mask = 110;

unsigned char or;

unsigned char and;

unsigned char not;

unsigned char leftOnce;

unsigned char leftTwice;

unsigned char rightOnce;

unsigned char rightTwice;

int i;

for(i=0;i<256;i++){

n = i;

or = n | mask;

and = n & mask;

not = ~n;

leftOnce = n << 1;

leftTwice = n << 2;

rightOnce = n >> 1;

rightTwice = n >> 2;

toBinary(n, result1);

toBinary(or, result2);

toBinary(and, result3);

toBinary(not, result4);

toBinary(leftOnce, result5);

toBinary(leftTwice, result6);

toBinary(rightOnce, result7);

toBinary(rightTwice, result8);

printf("%3hu %s %s %s %s %s %s %s %s\n",

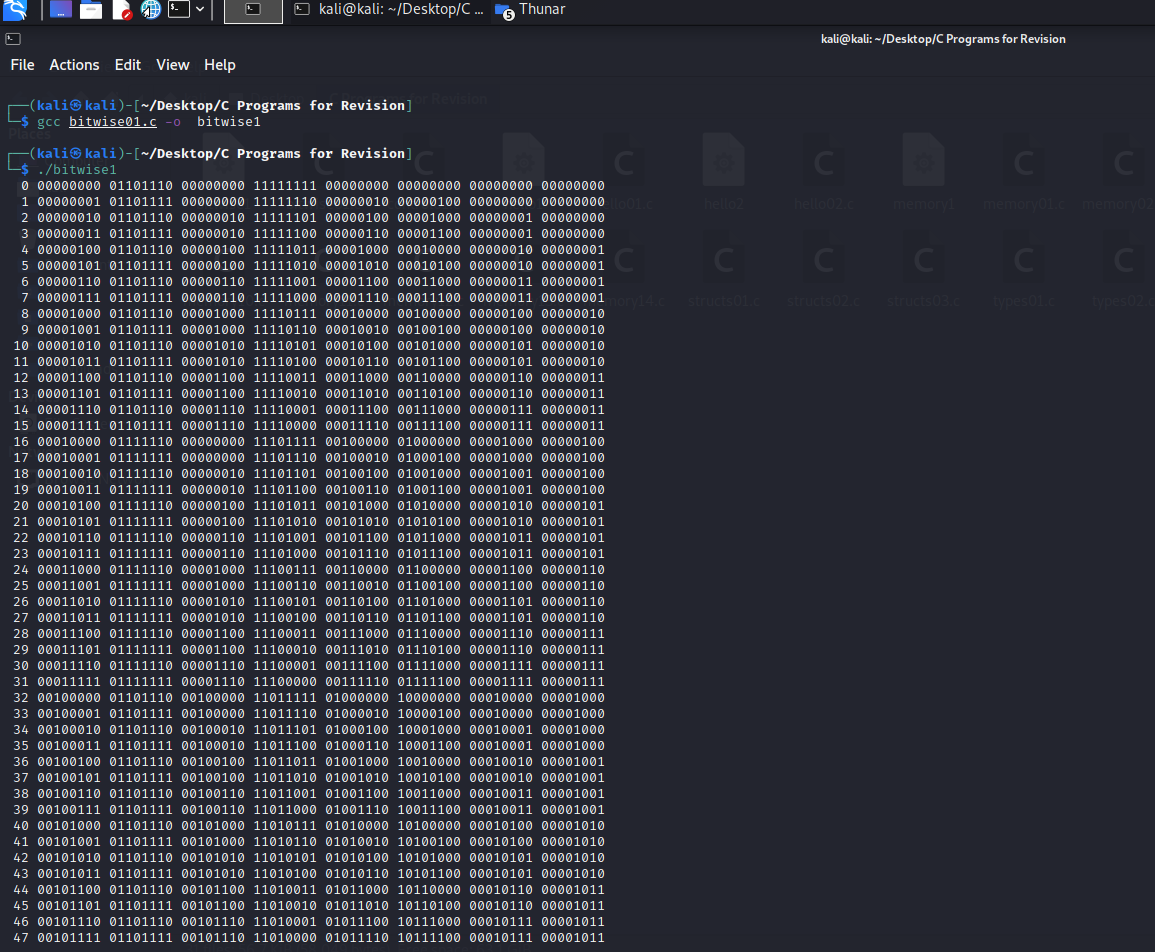
n, result1, result2, result3, result4,

result5, result6, result7, result8);

n++;

}

}



1. control01

#include <stdlib.h>

#include <stdio.h>

int main() {

int i;

for(i=0;i<5;i++){

printf("%d,", i);

}

printf("\n");

while(i<10){

printf("%d,", i);

i++;

}

do {

printf("%d,", i);

i++;

} while(i<15);

printf("\n");

if(i>13){

printf("custard\n");

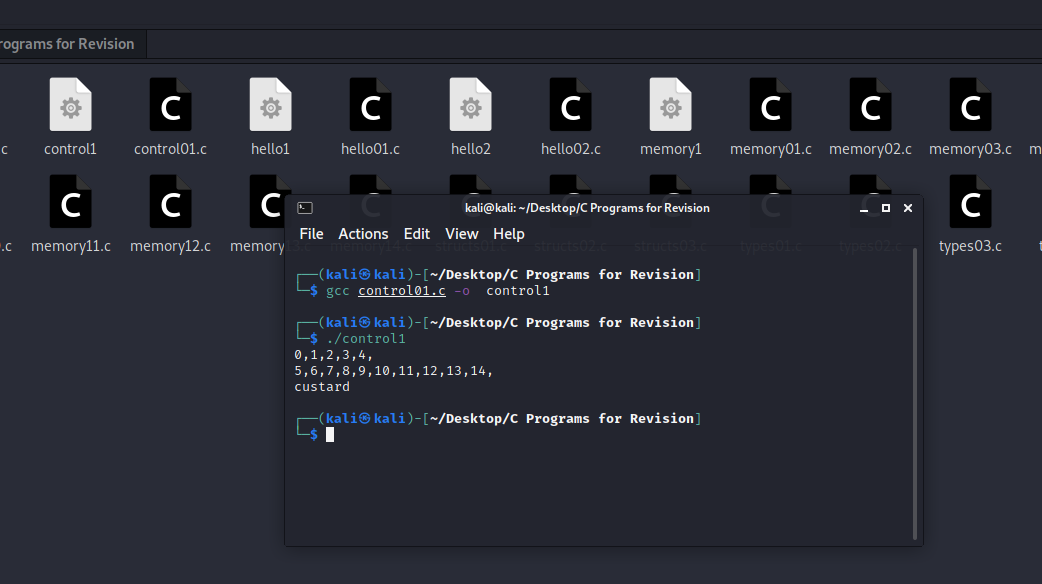
} else {

printf("gravy\n");

}

return EXIT\_SUCCESS;

}



1. hello01

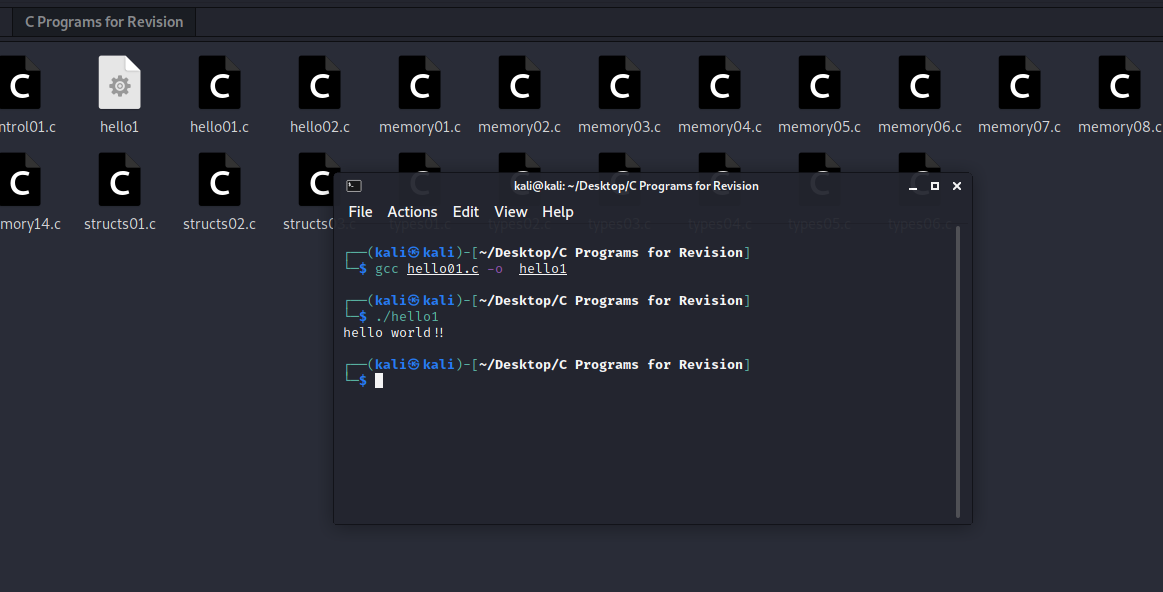
#include <stdio.h>

int main(){

printf("hello world!!\n");

return 0;

}



1. hello02

#include <stdlib.h>

#include <stdio.h>

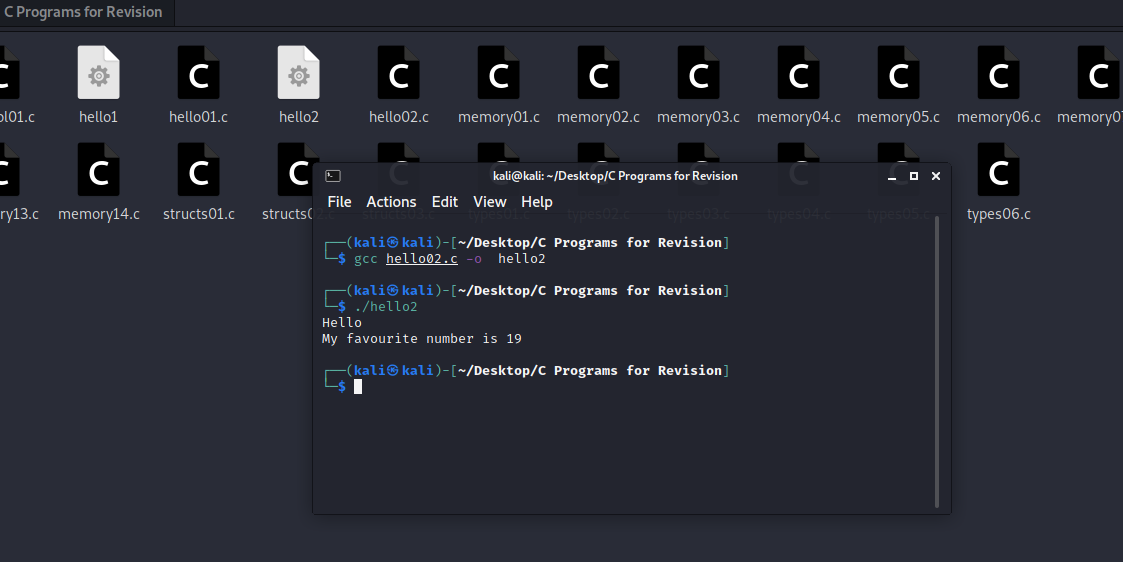
int main() {

int n = 19;

printf("Hello\nMy favourite number is %d\n", n);

return EXIT\_SUCCESS;

}



1. memory01

#include <stdio.h>

#include <stdlib.h>

int main() {

int x = 123;

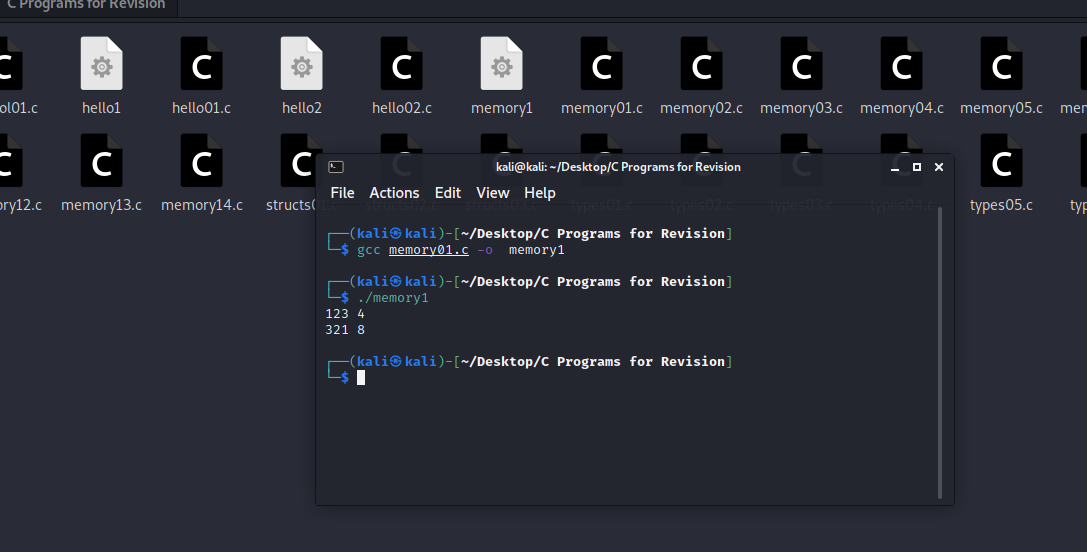
long int y = 321;

printf("%d %ld\n", x, sizeof(x));

printf("%ld %ld\n", y, sizeof(y));

return EXIT\_SUCCESS;

}



1. memory02

#include <stdio.h>

#include <stdlib.h>

int inc(int w) {

return w + 1;

}

int main() {

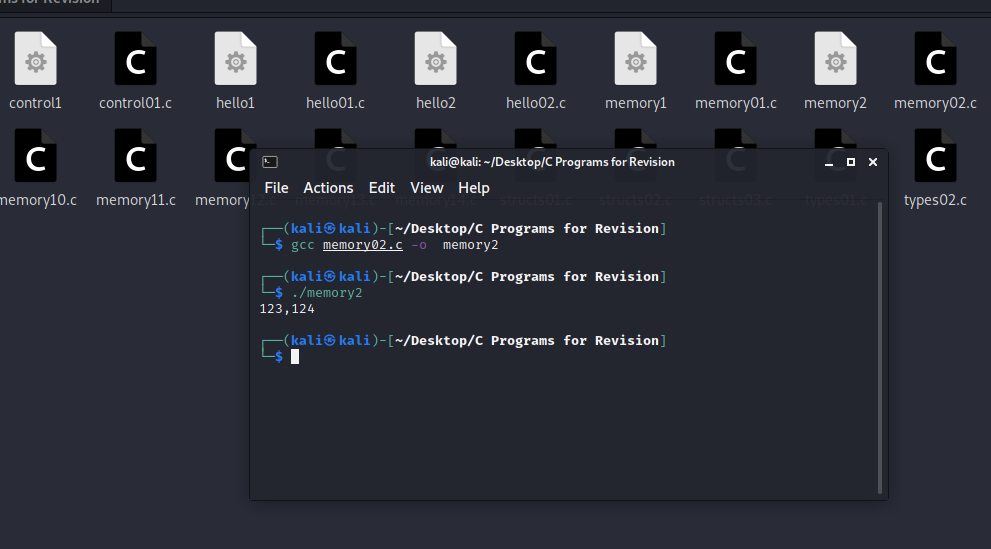
int x = 123;

int y = inc(x);

printf("%d,%d\n", x, y);

return EXIT\_SUCCESS;

}



1. memory03

#include <stdio.h>

#include <stdlib.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int x = 123;

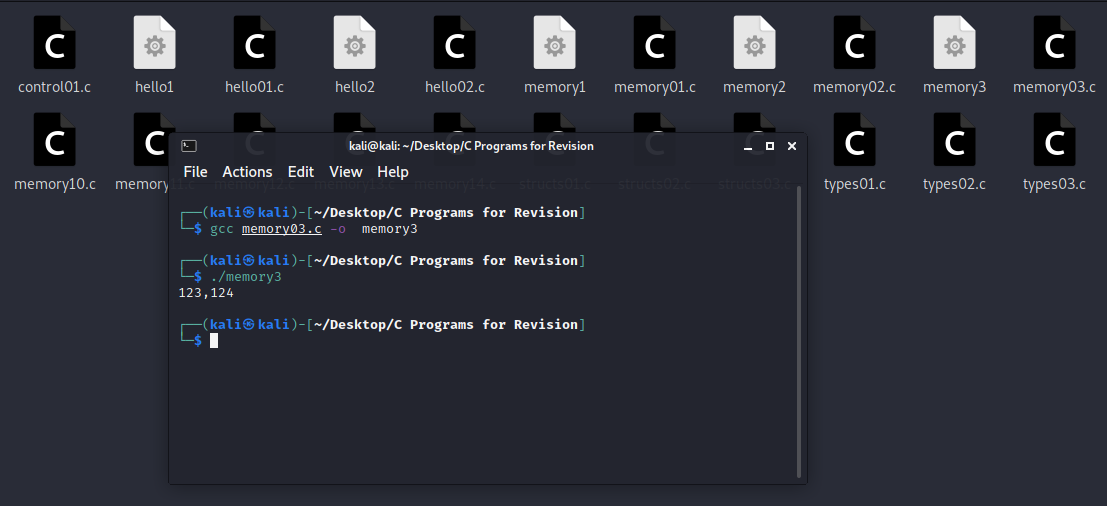
int y = x;

inc(&y);

printf("%d,%d\n", x, y);

return EXIT\_SUCCESS;

}



1. memory04

#include <stdio.h>

#include <stdlib.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int x = 123;

int \*y;

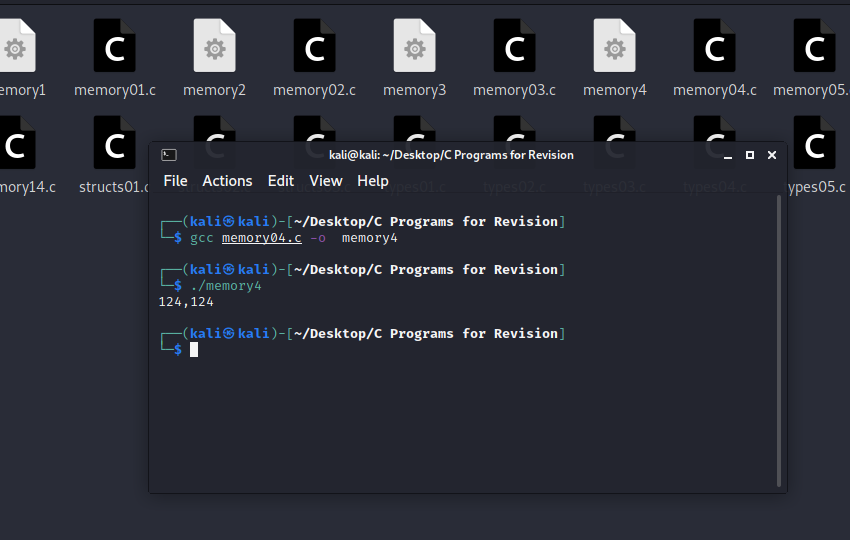
y = &x;

inc(y);

printf("%d,%d\n", x, \*y);

return EXIT\_SUCCESS;

}



1. memory05

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int \*x = malloc(sizeof(int));

\*x = 123;

int \*y;

y = x;

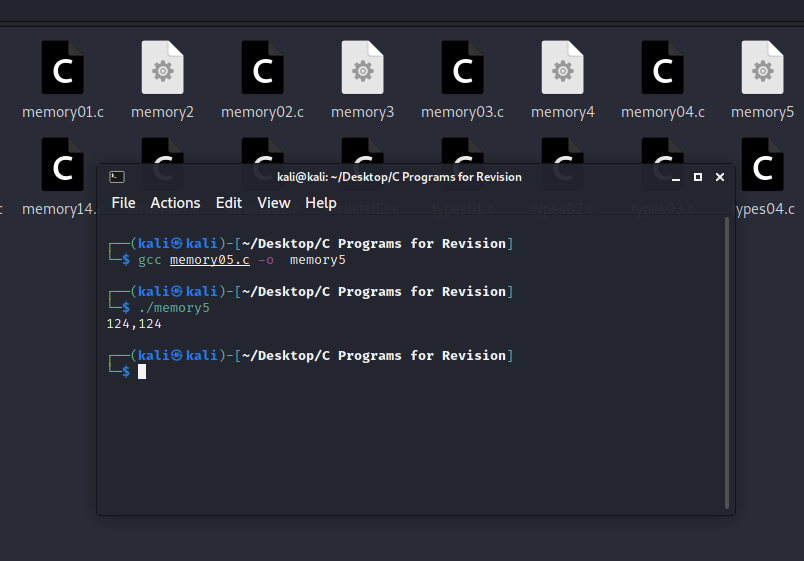
inc(y);

printf("%d,%d\n", \*x, \*y);

free(x);

return EXIT\_SUCCESS;

}



1. memory06

#include <stdio.h>

#include <stdlib.h>

struct pair {

int a;

int b;

};

int main() {

struct pair x;

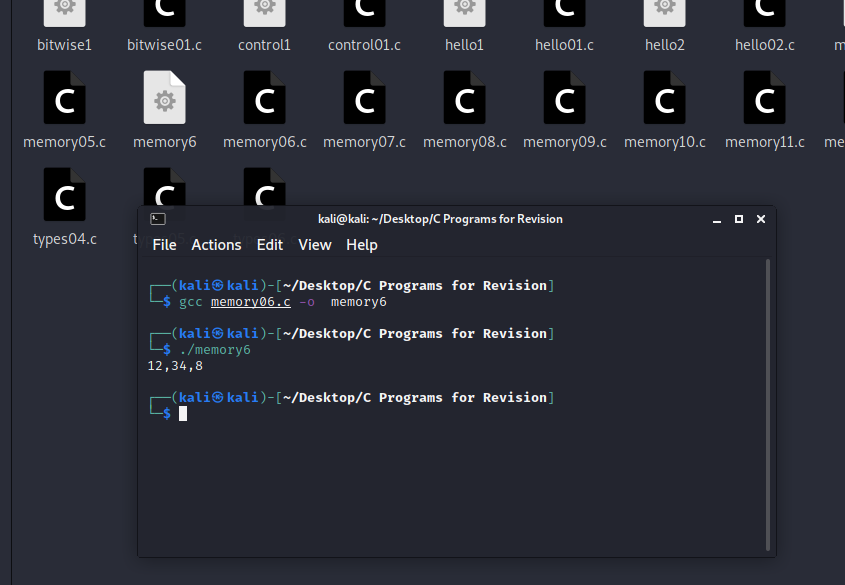
x.a = 12;

x.b = 34;

printf("%d,%d,%ld\n", x.a, x.b, sizeof(struct pair));

return EXIT\_SUCCESS;

}



1. memory07

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int a;

int b;

} pair;

int main() {

pair x;

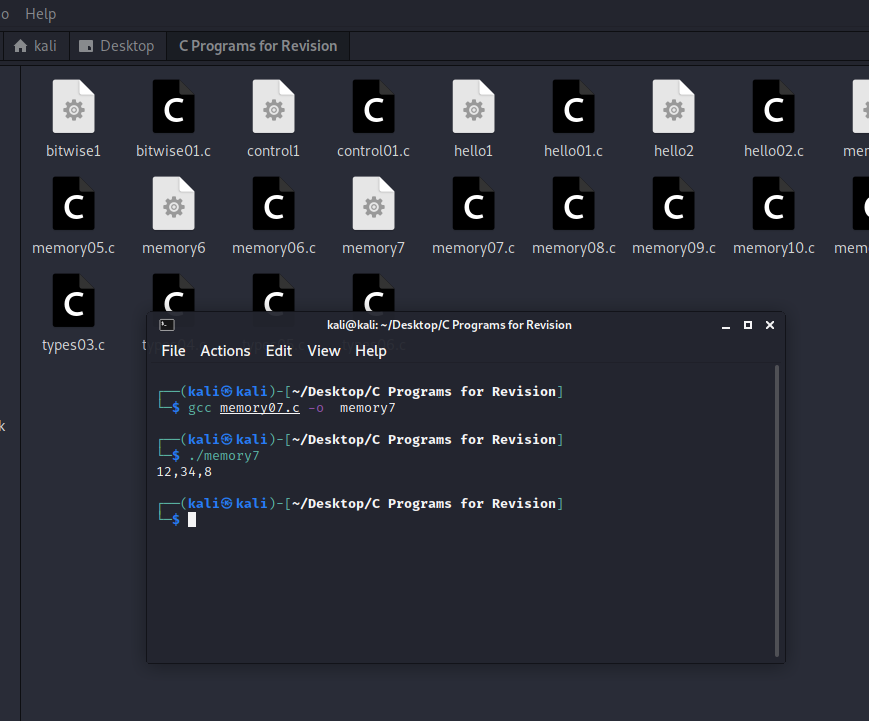
x.a = 12;

x.b = 34;

printf("%d,%d,%ld\n", x.a, x.b, sizeof(pair));

return EXIT\_SUCCESS;

}



1. memory08

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int a;

int b;

} pair;

void inc(pair \*w) {

w->a = w->a + 1;

w->b = w->b + 1;

}

int main() {

pair x;

x.a = 12;

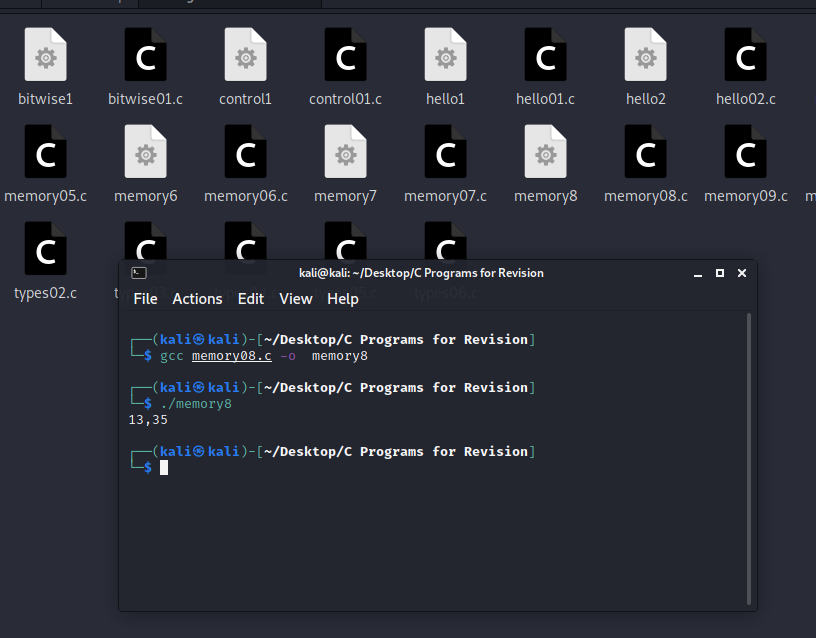
x.b = 34;

inc(&x);

printf("%d,%d\n", x.a, x.b);

return EXIT\_SUCCESS;

}



1. memory09

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

typedef struct {

int a;

int b;

} pair;

void inc(pair \*w) {

w->a = w->a + 1;

w->b = w->b + 1;

}

int main() {

pair \*x;

x = malloc(sizeof(pair));

x->a = 12;

x->b = 34;

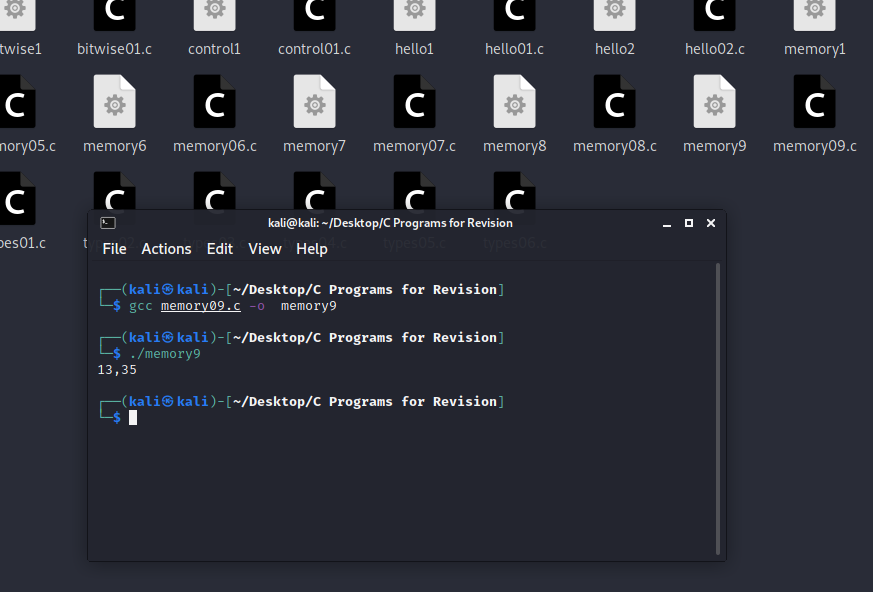
inc(x);

printf("%d,%d\n", x->a, x->b);

free(x);

return EXIT\_SUCCESS;

}



1. memory10

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

int main() {

int i;

int \*x;

x = malloc(sizeof(int) \* n);

printf("%ld\n", sizeof(x));

printf("%ld\n", sizeof(\*x));

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

x[i] = 2 \* i;

}

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

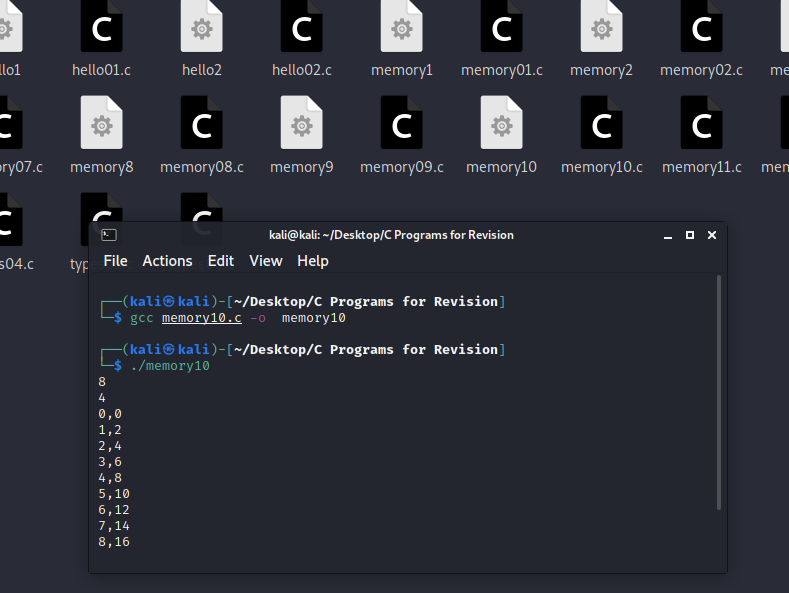
printf("%d,%d\n", i, x[i]);

}

free(x);

return EXIT\_SUCCESS;

}



1. memory11

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

y = x;

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

printf("%d,%d\n", i, \*y);

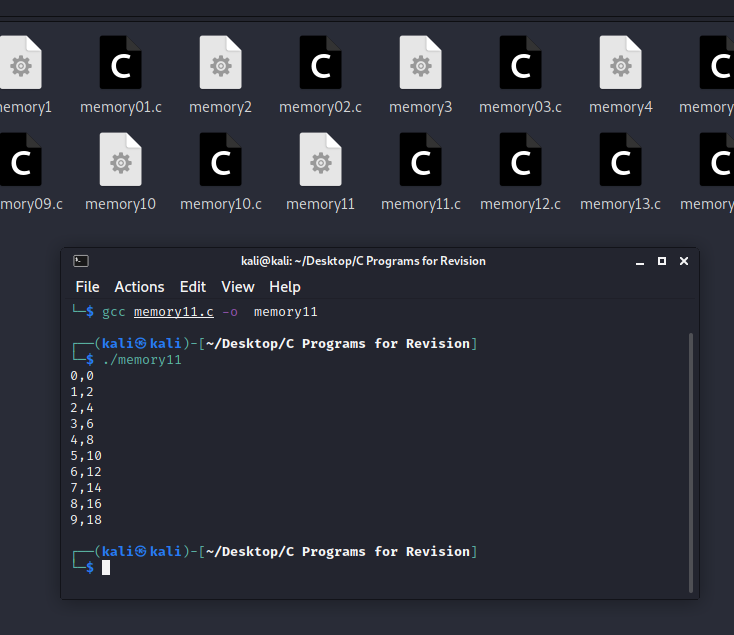
y++;

}

free(x);

return EXIT\_SUCCESS;

}



1. memory12

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void inc(int \*w) {

int i;

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

w[i] = w[i] + 1;

}

}

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

inc(x);

y = x;

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

printf("%d,%d\n", i, \*y);

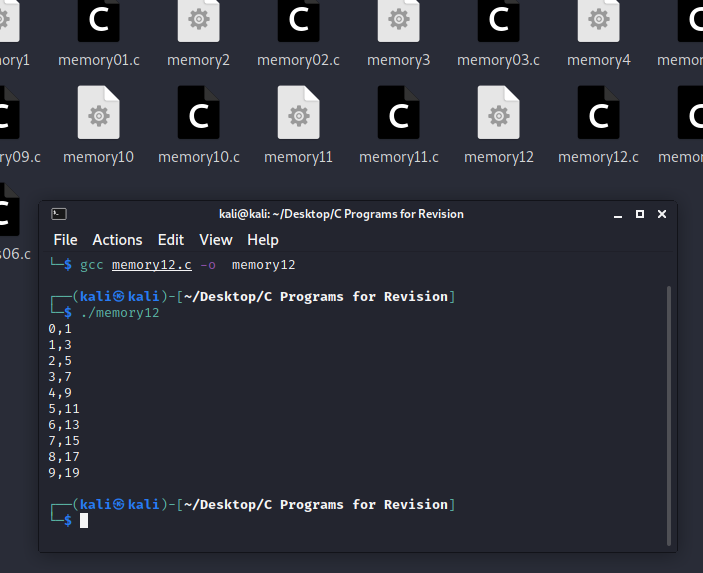
y++;

}

free(x);

return EXIT\_SUCCESS;

}



1. memory13

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void inc(int \*w) {

int i;

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

\*w = \*w + 1;

w++;

}

}

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

inc(x);

y = x;

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

printf("%d,%d\n", i, \*y);

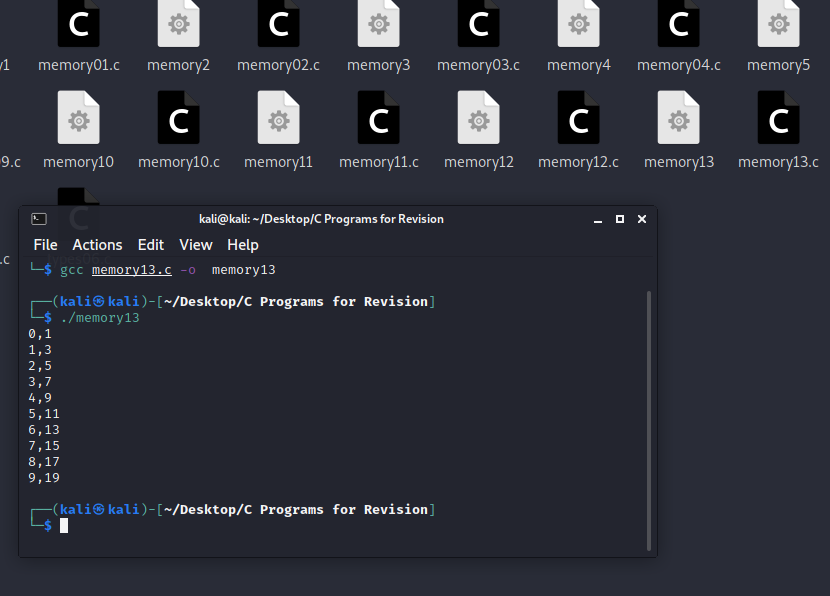
y++;

}

free(x);

return EXIT\_SUCCESS;

}



1. memory14

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void initialise(int \*w) {

int i;

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

\*w = 2 \* i;

w++;

}

}

void inc(int \*w) {

int i;

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

\*w = \*w + 1;

w++;

}

}

void output(int \*w) {

int i;

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

printf("%d,%d\n", i, w[i]);

}

}

int main() {

int \*x;

x = malloc(sizeof(int) \* n);

initialise(x);

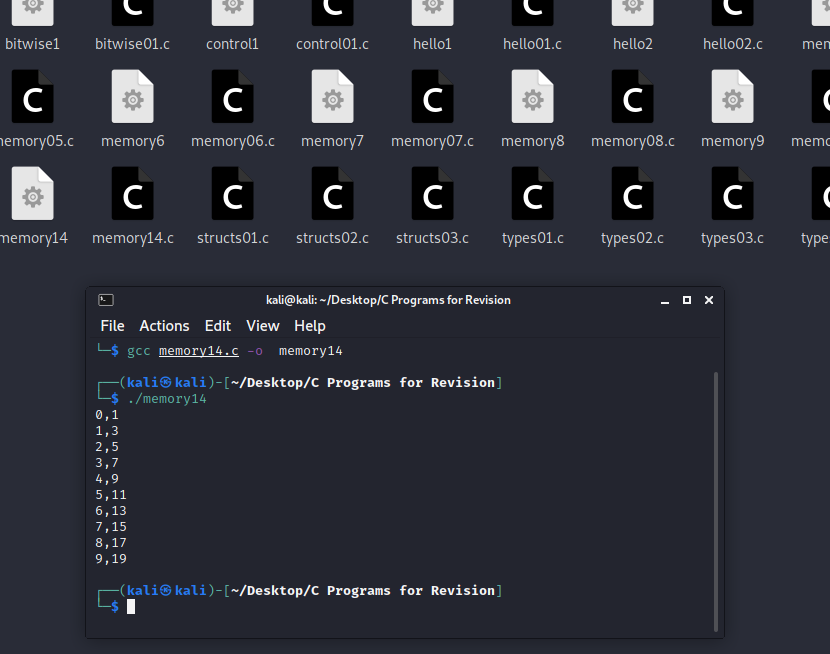
inc(x);

output(x);

free(x);

return EXIT\_SUCCESS;

}



1. structs01

#include <stdio.h>

struct t {

unsigned int h;

unsigned int m;

unsigned int s;

};

int main() {

struct t a;

struct t \*b;

a.h = 5;

a.m = 9;

a.s = 45;

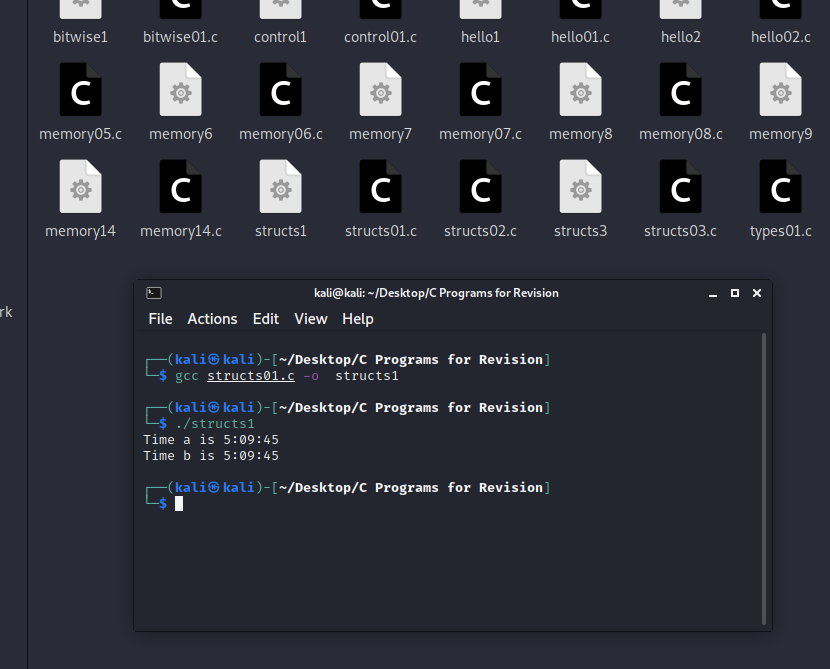
printf("Time a is %u:%02u:%02u\n", a.h, a.m, a.s);

b = &a;

printf("Time b is %u:%02u:%02u\n", b->h, b->m, b->s);

return 0;

}



1. structs02

#include <stdio.h>

typedef struct {

unsigned int h;

unsigned int m;

unsigned int s;

} t;

int main() {

t a;

t \*b;

a.h = 5;

a.m = 9;

a.s = 45;

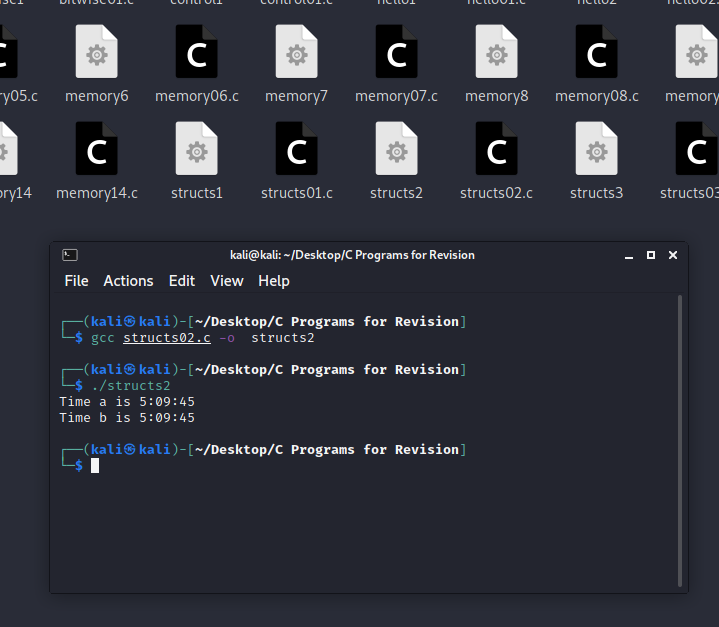
printf("Time a is %u:%02u:%02u\n", a.h, a.m, a.s);

b = &a;

printf("Time b is %u:%02u:%02u\n", b->h, b->m, b->s);

return 0;

}



1. structs03

#include <stdio.h>

#include <malloc.h>

typedef struct {

unsigned int h;

unsigned int m;

unsigned int s;

} t;

int main() {

t \*a;

a = (t \*) malloc(sizeof(t));

a->h = 5;

a->m = 9;

a->s = 45;

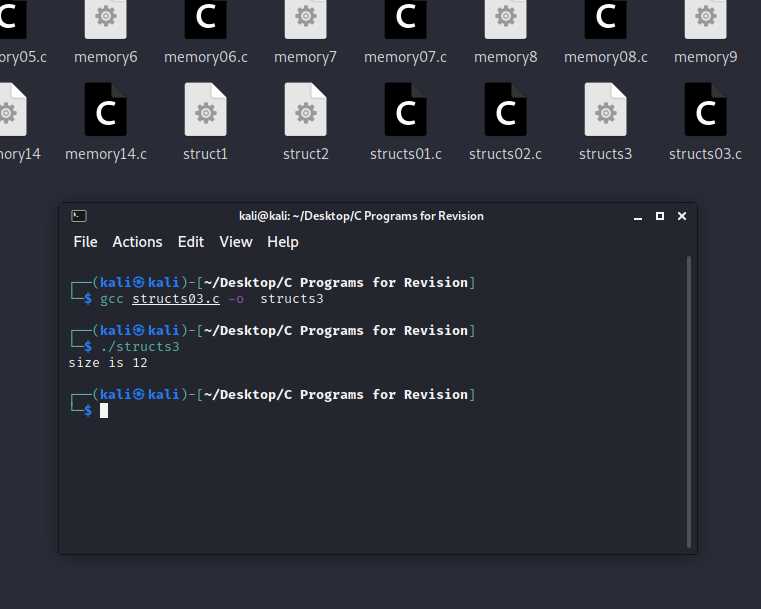
//printf("Time a is %u:%02u:%02u\n", a->h, a->m, a->s);

printf("size is %d", sizeof(t));

//free(a);

return 0;

}



1. types01

#include <stdlib.h>

#include <stdio.h>

int main() {

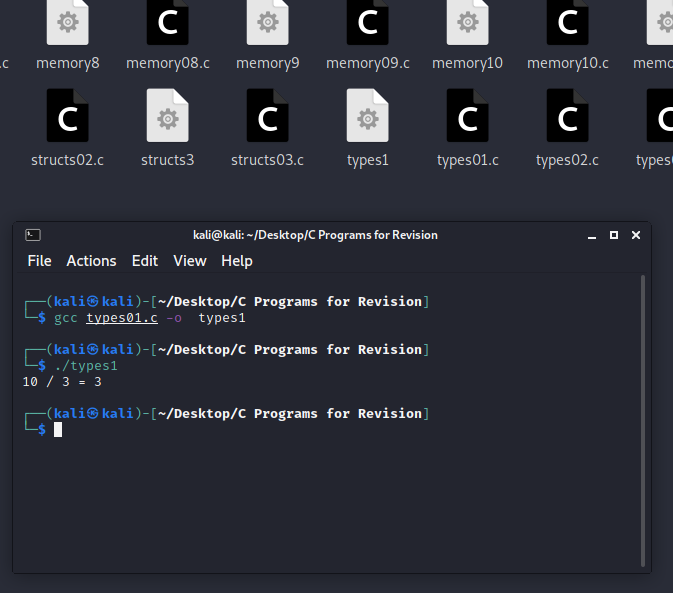
int x = 10;

int y = 3;

printf("%d / %d = %d\n", x, y, x/y);

return EXIT\_SUCCESS;

}



1. types02

#include <stdlib.h>

#include <stdio.h>

int main() {

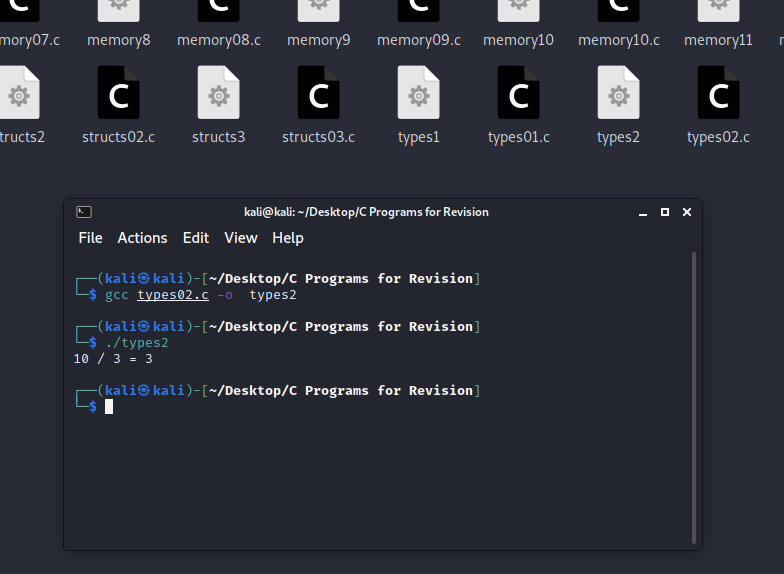
long int x = 10L;

long int y = 3L;

printf("%ld / %ld = %ld\n", x, y, x/y);

return EXIT\_SUCCESS;

}



1. types03

#include <stdlib.h>

#include <stdio.h>

int main() {

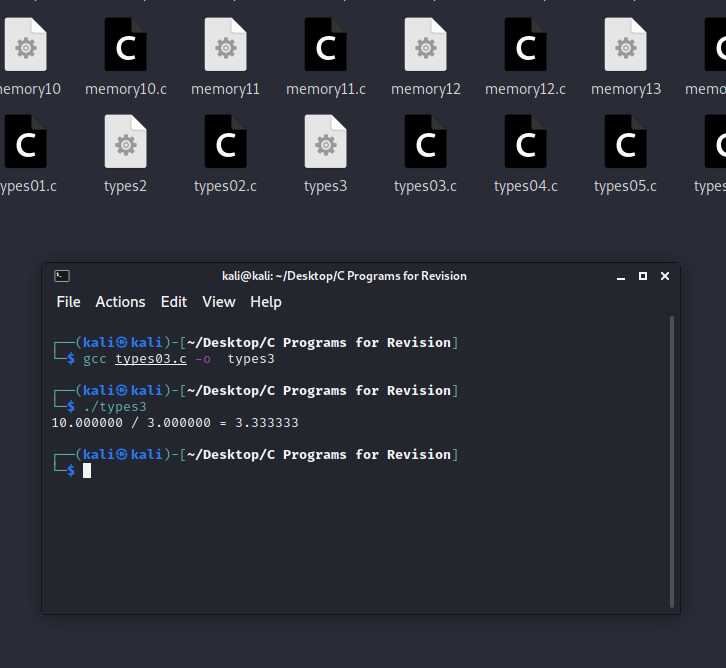
float x = 10.0f;

float y = 3.0f;

printf("%f / %f = %f\n", x, y, x/y);

return EXIT\_SUCCESS;

}



1. types04

#include <stdlib.h>

#include <stdio.h>

int main() {

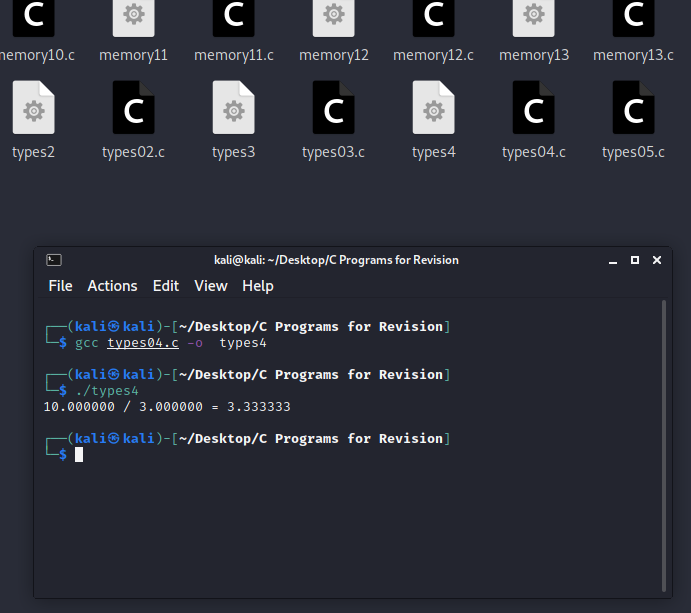
double x = 10.0;

double y = 3.0;

printf("%lf / %lf = %lf\n", x, y, x/y);

return EXIT\_SUCCESS;

}



1. types05

#include <stdlib.h>

#include <stdio.h>

int main() {

int a = 2;

int b = 3;

int c = 2;

int d = 4;

printf("There are no booleans in c\n");

printf("%d\n", a==b);

printf("%d\n", a==c);

printf("%d\n", a!=b);

printf("%d\n", a!=c);

printf("%d\n", a==b);

printf("%d\n", !(a==b));

int e = (a == b) || (a == c);

int f = (a == b) && (a == c);

printf("e=%d\n", e);

printf("f=%d\n", f);

if(e) {

printf("e=true\n");

} else {

printf("e=false\n");

}

if(f) {

printf("f=true\n");

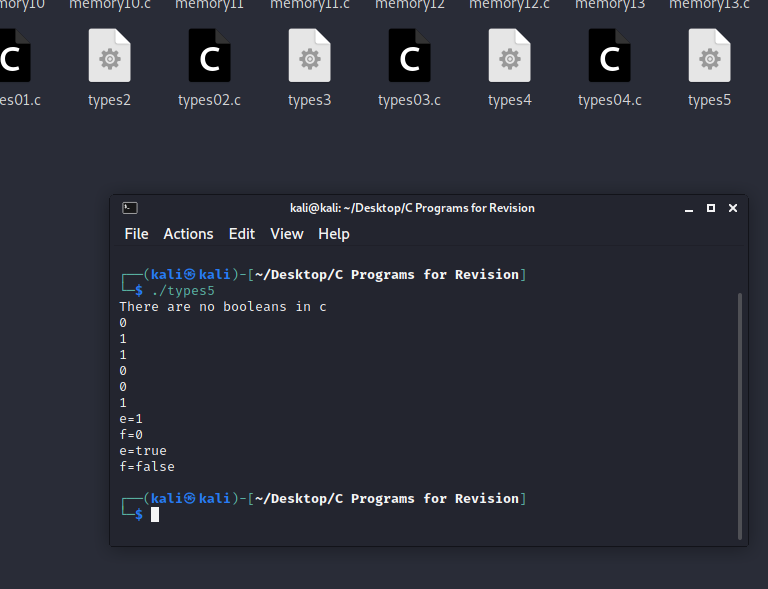
} else {

printf("f=false\n");

}

return EXIT\_SUCCESS;

}



1. types06

#include <stdlib.h>

#include <stdio.h>

int main() {

printf("Strings are just arrays of chars\n");

char \*message1 = "hello";

char \*message2 = "kevan";

printf("%s %s\n", message1, message2);

printf("Look in /usr/include/string.h for functions\n");

printf("that can be applied. Each has a man page.\n");

return EXIT\_SUCCESS;

}

