//Tanveer Bariana

//csc 60

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

//ptr0.c

#include <stdio.h>

int main() {

int i, j, \*p, \*q, \*\*r;

printf("Make sure it's 4 (bytes): size of int \* is %i\n",sizeof(int \*));

printf("addr: %u %u %u %u %u\n", &i, &j, &p, &q, &r);

// give p an address (p then points there),

// / \*p = 33 or \*p = i are illegal, only p = 0 is valid to denote null

i = 33; // i = 33

p = &i; // p = 4292380364

printf(" \*p is %u\n", \*p);

printf(" 2\*\*P is %u\n", 2\*\*p);

printf(" p is %u\n", p);

printf("&i is %u\n", &i);

printf("&p is %u\n", &p);

j = (int)p; //j= 4292380364

printf("j is %i\n", j);

printf(" \*(int \*)j is %i\n", \*(int \*)j);

\*p = 44; //i=44

printf("\*p is now %u\n", \*p);

printf("p is now %u\n", p); // print p ?

printf("&p is now %u\n", &p); // print &p ?

i = 5; // i= 5

j = 7; // j=7

q = p; // q=4292380364

printf("&i is now %u\n", &i);

printf("q is now %u\n", q); // print q ?

printf("&q is now %u\n", &q); // print &q ?

printf("\*q is %u at last\n", \*q); // print \*q ?

r = &q; // r=4292380352

printf("r is now %u\n",r); // print r ?

printf("&r is now %u\n",&r); // print &r ? //

r = &q; // r=4292380352

printf("r is now %u\n",r); // print r ?

printf("&r is now %u\n",&r); // print &r ?

printf("\*r is now %u\n",\*r); // print \*r ?

printf("\*\*r is now %u\n",\*\*r); // print \*\*r ?

\*r = &j; //q=4292380360

printf("r is finally %u\n",r); // print r ?

printf("&r is finally %u\n",&r);// print &r ?

printf("\*r is finally %u\n",\*r); // print \*r ?

printf("\*\*r is finally %u\n",\*\*r); // print \*\*r ?

printf("lastly &j is %u\n",r); // print &j ?

printf("lastly q is %u\n",q); // print q ?

printf("lastly &q is %u\n",&q); // print &q ?

printf("lastly \*q is %u\n",\*q); // print \*q ?

return 0;

}

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

//ptr1.c

#include <stdio.h>

#include <stdlib.h>

#define N 10 // size of array (why fixed?)

void Sub1(int []);

void Sub2(int \*);

int main() {

int y[1]; // int \*y; y = (int \*)malloc(sizeof(int));

int i;

int x[] = {40, 51, 62, 73, 84, 95, 106, 117, 128, 139};

int \*p, \*q;

printf("0. %i\n", sizeof(int [1]));

printf("1. %d\n", sizeof(int [N]));

p = x; // p = 4286619288. addr of x[0]

printf("ps is %u.\n", p);

q = p; // q = 4286619288.

printf("2. \*x is %d, x is %u\n", \*x, x);

printf("3. value of q is %u, as addr q points at %d\n", q, \*q);

printf(" &x[0] is %d\n", &x[0]);

q = &x[0]; //q= -8348008

\*q = 10; //x[0] = 10

printf("4. value %d x+1 is %u\n", \*(x+1), x+1);

printf("5. value %d p+1 is %u\n", \*(p+1), p+1);

printf("6. value %d q+1 is %u\n", \*(q+1), q+1);

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

printf("7. main: value %d in addr %u\n", q[i], q+i);

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

printf("8. main: value %d in addr %u\n", x[i], x+i);

Sub1(x);

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

printf("9. main: value %d in addr %u\n", \*(x+i), x+i);

Sub2(x);

return 0;

}

void Sub1(int y[N]) {

int j;

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

printf("A. Sub1: value %i in addr %u\n", \*(y+j), y+j);

y[0] = 999; //

}

void Sub2(int \*z) {

int k;

\*(z+3) = 222;

for(k=0; k<N; k++)

printf("B. Sub2: value %i in addr %u\n", \*(z+k), z+k);

printf("C. z[3] is %d\n", z[3]);

}

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

//ptr2.c

#include <stdio.h>

ChangeX(int a) { a = 500; }

Change0(int \*p) { \*p = 2; } // can change the value in the addr of what p has

Change1(int \*p) { int i = 10; p = &i; } // can't change the addr p points to to i's

Change2(int \*\*dog) { int i = 20; \*dog = &i; } // unless p's addr is passed over

Change3(int \*p) { \*p = 30; } // can change value in the addr of what p has

Change4(int \*p) { int i = 40; p = &i; } // can't change addr p points to to i's

Change5(int \*\*p) { int i = 50; \*p = &i; } // unless p's addr is passed over

int main() {

int i, j, \*p;

i = 1; j = 2; p = &i;

printf("Before: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 1 j = 2 p = 4287273324 \*p = 1

ChangeX( i );

printf("After ChangeX: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 500 j = 2 p = 4287273324 \*p = 1

Change0( p );

printf("After Change0: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 2 j = 2 p =4287273324 \*p = 2

Change1( p );

printf("After Change1: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 10 j = 2 p = 4287273324 \*p = 2

Change2( &p );

printf("After Change2: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 20 j = 2 p = 4287273284 \*p = 20

j=3; p=&j;

printf("Before: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 20 j = 3 p = 4287273320 \*p = 3

Change3( p );

printf("After Change3: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 20 j = 30 p = 4287273320 \*p = 30

Change4( p );

printf("After Change4: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 40 j = 30 p = =4287273320 \*p = 30

Change5( &p );

printf("After Change5: \*p=%u p=%u &p=%u\n\n", \*p, p, &p);

// i = 50 j = 30 p = 4287273284 \*p = 50

return 0;

}