Prelab 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Pre 1.1

a. No. In pass by value, copies of the values of the are passed to the function.

b.

void bar(int \*myVar);

c.

void bar(int \*myVar) {

\*myVar = 42;

}

Yes, we are using a pointer to point myVar. It is passing by reference.

d. Comparing with “pass by value” and “pass by reference”, “pass by value” is passing a copy of the variable to a function (which doesn’t change the original variable).

But “pass by reference” is passing an alias of the variable to a function (which is changing the variable).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Pre 1.2

a. Decimal integer

Printf(%d, myVar)

or

Printf(%i, myVar)

b. Hexadecimal integer

Printf(%x, myVar)

c. Octal value

Printf(%o, myVar)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Pre 1.3

#include <stdio.h>

/\* function takes two integers and sum them together \*/

int add(int first, int second) {

return first + second;

}

/\* function takes two integers and substract the second from first integers \*/

int substract(int first, int second) {

return first - second;

}

/\* function takes two integers and mutiplys them \*/

int multiply(int first, int second) {

return first \* second;

}

/\* function takes two integers and returns the result first integer is diveded by the second \*/

int divide(int first, int second) {

return first / second;

}

/\* the main function \*/

int main(void) {

/\*scan the input of positive integers\*/

printf("Enter 2 positive integers for calculation:\n");

printf("Warning: positive integers only, other inputs may crash the program\n");

int n, m;

scanf("%d%d", &n, &m);

/\*check the input integers are positive\*/

if (n <= 0 || m <= 0) {

printf("Your inputs are invalid, please reenter 2 positive integers\n");

scanf("%d%d", &n, &m);

}

printf("Please choose an operation from (+, -, \*, /)");

printf("Warning: operation only, other inputs may crash the program\n");

/\*scan the operation to call the coresponding function\*/

char s;

scanf("%s", &s);

if (s == '+') {

printf("%i+%i = %i\n", n, m, add(n, m));

}

else if (s == '-') {

printf("%i-%i = %i\n", n, m, substract(n, m));

}

else if (s == '\*') {

printf("%i\*%i = %i\n", n, m, multiply(n, m));

}

else if (s == '/') {

printf("%i/%i = %i\n", n, m, divide(n, m));

}

/\*print error if user entered invalid inputs\*/

else {

printf("Your input was invalid, please restart the program\n");

}

return 0;

}