**ENEE140 “Quiz”**

*Multiple Choice (3 minutes)*

1. What will be the output of the following C code?

#include <stdio.h>

int main() {

int x = 5;

int y = 7;

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

return 0;

}

a) 42

b) 36

c) 35

d) 30

1. Which of the following is the correct syntax to declare a function pointer in C?

a) int (\*ptr)();`

b) int ptr();`

c) int \*ptr();`

d) int (\*ptr)[];`

1. What will be the output of the following C code?

#include <stdio.h>

int main() {

int i = 5;

while (--i) {

printf("%d ", i);

}

return 0;

}

a) 5 4 3 2

b) 4 3 2 1

c) 5 4 3 2 1

d) 4 3 2

1. In C, which function is used to open a file for writing?

a) fopen()

b) open()

c) write()

d) fopen\_s()

1. What will be the output of the following C code?

#include <stdio.h>

void mystery(int n) {

if (n > 1) {

mystery(n / 2);

printf("%d ", n % 2);

}

}

int main() {

mystery(7);

return 0;

}

a) 1 1 1

b) 0 1 0 1

c) 1 1 1 0

d) 1 0 1 1

1. Which of the following is the correct syntax to close a file in C?

a) closefile(fp);`

b) close(fp);`

c) fclose(fp);`

d) fclosefile(fp);`

*Free Response (25-30 minutes)*

1. Find the bugs in the code:

#include <stdio.h>

int main() {

int i, j;

int sum = 0;

// Initialize array

int arr[5] = {1, 2, 3, 4, 5};

// Calculate sum of elements

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

sum += arr[i];

}

printf("Sum of array elements: %d\n", sum);

// Check for prime numbers

for (i = 2; i <= 10; i++) {

for (j = 2; j <= i/2; j++) {

if (i % j == 0) {

printf("%d is not a prime number\n", i);

break;

}

}

if (j > i/2) {

printf("%d is a prime number\n", i);

}

}

return 0;

}

1. Convert this loop:  
   #include <stdio.h>

int main() {

int i = 0;

int sum = 0;

// Convert the following while loop to a do-while loop

while (i < 10) {

sum += i;

i++;

}

printf("Sum: %d\n", sum);

return 0;

}

1. Given this code, determine the output:

#include <stdio.h>

int main() {

int x = 10;

int y = 5;

int z = 0;

if (x > 5) {

if (y > 2) {

z = x + y;

} else {

z = x - y;

}

} else if (x < 5) {

z = x \* y;

} else {

z = x / y;

}

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

return 0;

}

1. BONUS (may be worth some bonus points, for real): You may have forgotten about CLONEWARS, but CLONEWARS never leaves you. Find out what the CLONEWARS became:

#include <stdio.h>

float NEWEMPIRE(float galactic);

int REBEL(float alliance);

void FORCE(int jedi);

int main() {

int CLONEWARS = 0;

int i;

for (i = 1; i <= 10; i++) {

CLONEWARS += (i \* i \* i) - (i + 10);

}

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

CLONEWARS = NEWEMPIRE((float)CLONEWARS);

printf("CLONEWARS = %.1f\n", (float)CLONEWARS);

CLONEWARS += (35 \* 2) - (7 \* 3) + 11;

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

FORCE(CLONEWARS);

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

CLONEWARS += 0xAB;

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

CLONEWARS = REBEL(CLONEWARS);

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

return 0;

}

float NEWEMPIRE(float galactic) {

int starDestroyer;

float executor = 0.0;

for (starDestroyer = 0; starDestroyer < 15; starDestroyer++) {

executor = starDestroyer \* (1.2 \* executor);

galactic -= executor / 7;

}

if (galactic >= 250) {

return galactic \* 3.0;

} else {

return executor + 12.0;

}

}

void FORCE(int jedi) {

int sith = 0;

do {

sith += jedi \* 2;

jedi = sith % 20;

} while (sith % jedi != 0);

}

int REBEL(float alliance) {

int xWing;

int falcon = 0;

for (xWing = 5; xWing > 0; xWing--) {

falcon = (xWing \* 3) + 1;

}

if (falcon > 10) {

alliance = 250;

return alliance;

} else {

alliance = 230;

return alliance;

}

}

1. BONUS (may be worth some bonus points, for real): Given a lengthy binary number (it can be broken down), give me the Octal, Hex, Decimal, and ASCII conversions (once you get it, turn it in to me):

01000110 01001111 01010010 00100000 01010100 01001000 01000101 00100000 01010010 01000101 01010000 01010101 01000010 01001100 01001001 01000011 00100001 00100001 00100001 00100001

Answers:

MCQ:

1. 35
2. int (\*ptr)();
3. 4 3 2 1
4. fopen()
5. 1 0 1 1
6. fclose(fp);

FRQ:

1. See bugs:
   1. In the first loop (for (i = 0; i <= 5; i++)), the loop condition should be i < 5 instead of i <= 5, to prevent accessing elements beyond the array bounds
   2. In the second loop (prime number check), the inner loop condition should be j <= i / 2 instead of j <= i / 2, to properly check for prime numbers

Corrected Code:

#include <stdio.h>

int main() {

int i, j;

int sum = 0;

// Initialize array

int arr[5] = {1, 2, 3, 4, 5};

// Calculate sum of elements

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

sum += arr[i];

}

printf("Sum of array elements: %d\n", sum);

// Check for prime numbers

for (i = 2; i <= 10; i++) {

for (j = 2; j <= i/2; j++) {

if (i % j == 0) {

break;

}

}

if (j > i/2) {

printf("%d is a prime number\n", i);

} else {

printf("%d is not a prime number\n", i);

}

}

return 0;

}

1. #include <stdio.h>

int main() {

int i = 0;

int sum = 0;

// Convert the following while loop to a do-while loop

do {

sum += i;

i++;

} while (i < 10);

printf("Sum: %d\n", sum);

return 0;

}

1. z = 15
2. CLONEWARS = 24

CLONEWARS = 37.3

CLONEWARS = 63

CLONEWARS = 13

CLONEWARS = 217

CLONEWARS = 250

1. FOR THE REPUBLIC!!!!