#  Repetition Homework

1. What output does the following program fragment produce?

i = 4;

while (i <= 128) {

printf("%d ", i);

i \*= 2;

}

4 8 16 32 64 128

2. What output does the following program fragment produce?

i = 4321;

do {

printf("%d ", i);

i /= 10;

} while (i > 0);

4321 432 43 4

3. Show the output of each of the code blocks below:

a) for (i = 5; i < 32; i += 6) {

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

}

5

11

17

23

29

b) for (i = 25; i > 0; i -= 5) {

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

}

25

20

15

10

5

4. Write C statements to sum the odd integers from 1 and 100 using a for statement. Assume the integer variables sum and count have been defined, but not initialized, to be of type int.

sum = 0;

for ( count = 1; count <= 100; count += 2 )

sum += count;

5. Write for statements that print the following sequences of values:

1. 5, 10, 15, 20, 25

for ( i = 10; i <= 50; i += 10 )

printf("%d ", i );

b) 75, 65, 55, 45, 35

for ( i = 75; i >= 35; i -= 10 )

printf("%d ", i );

6. Assume that we want to calculate the sum of the integers from 50 down to 1. What is wrong with the code below?

sum = 0;

num = 50;

while (num >= 0) {

sum += num;

}

num should be decremented in the while loop.

7. See if you can improve the following program segments through reorganization. First analyze the code to determine the function actually being performed. Then recode that function as clearly and simply as possibly (i.e. without the use of break or continue).

a) while(a) { b) do {

if(b) { if (!a) {

continue; continue;

} }

c(); else {

} b();

c();

}

} while (a);

a) Note that the need for a continue can often be eliminated by altering a condition:

while (a)

if (!b)

c();

b) Once again we can eliminate the continue construct:

do {

if (a) {

b();

c();

}

}

while (a);

Then we can replace the do...while construct with a while loop:

while (a) {

b();

c();

}