ASSIGNMENT: 06

NAME: SYEDA FAIZA ASLAM

ROLL NO: CT-25064

SECTION: B

COURSE: CODE IN PLACE

TASKS:

QUESTION NO: 01

– Fibonacci Challenge.

Write a program to print Fibonacci numbers up to

500.

Skip numbers divisible by 5.

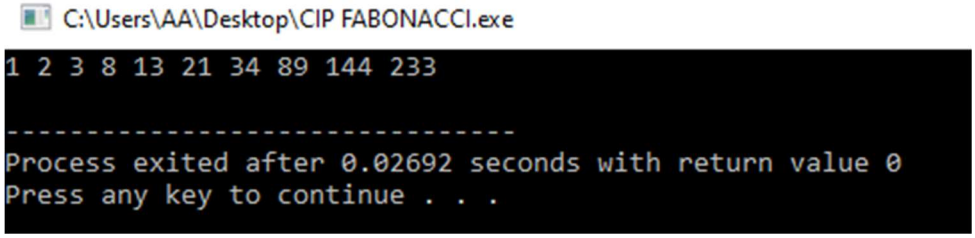
Stop if a number becomes greater than 300.

(Hint: Use for or while, along with continue and break.)

SOURCECODE:

```
#include <stdio.h>

int main() {
   int a = 0, b = 1, next;

   while (1) {
      next = a + b;
      // Stop if number becomes greater than 300
      if (next > 300) {
         break;
      }

      // Skip numbers divisible by 5
      if (next % 5 != 0) {
         printf("%d ", next);
      }

      a = b;
      b = next;
   }
   printf("\n");
   return 0;
}
```

OUTPUT:



QUESTION NO: 02

Student MarksTracker

Keep taking marks from user until -1 is entered.

Count how many are passing (≥50) and failing (<50).

If more than 3 fails are entered in a row → stop input automatically.

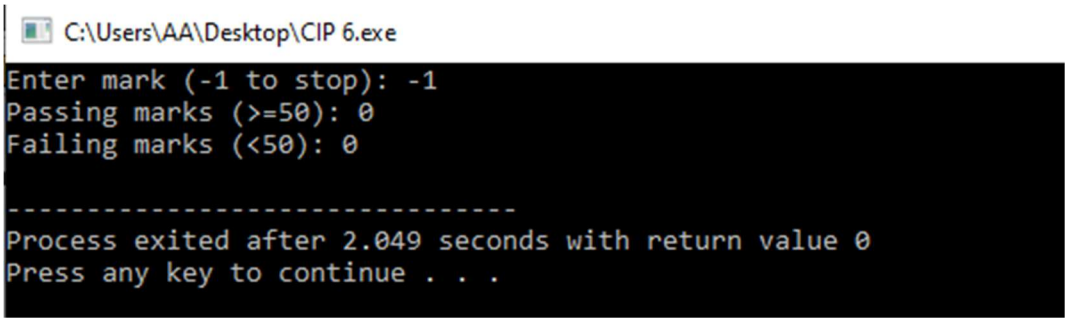
(Hint: Use while + condition + break.)

SOURCECODE:

```
#include <stdio.h>

int main() {
   int mark, passCount = 0, failCount = 0, consecutiveFails = 0;

   while (1) {
      printf("Enter mark (-1 to stop): ");
      scanf("%d", &mark);

      if (mark == -1) {
         break;
      }

      if (mark >= 50) {
         passCount++;
         consecutiveFails = 0; // Reset consecutive fails on pass
      } else {
         failCount++;
         consecutiveFails++;
      }

      // Stop input if more than 3 fails in a row
      if (consecutiveFails > 3) {
         printf("More than 3 fails in a row. Stopping input.\n");
         break;
      }
   }

   printf("Passing marks (>=50): %d\n", passCount);
   printf("Failing marks (<50): %d\n", failCount);
```

```
    return 0;
}
```

OUTPUT:



Q U I Z :

Iterative Structures in C

Multiple Choice Questions (MCQs)

1. Which loop is guaranteed to execute at least once?

a) for loop

b) while loop

c) do-while loop

d) nested loop

2. What will be the output of the following code?

```
int i = 3;
while(++i < 7) {
printf("%d ", i);
}
```

a) 3 4 5 6

b) 4 5 6

c) 4 5 6 7

d) 5 6 7

3. In a for loop for(init; condition; increment), when is the increment statement

executed?

a) Before the condition is checked

b) After the loop body is executed

c) Before the loop body is executed

d) Only when the condition is false

4. Which statement is used to skip the current iteration and continue with the next

iteration?

a) break

b) continue

c) return

d) goto

5. What is the correct syntax for a nested for loop?

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

for(j=0; j<3; j++)

{ statements; }

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

{

for(j=0; j<3; j++)

{ statements; }

}

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

for(j=0; j<3; j++)

{ statements; }

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

(j=0; j<3; j++)

{ statements; }

6. The condition in a while loop is checked before executing the loop body.

True / False

7. A break statement inside a nested loop will terminate all the loops.

True / False

8. The following code will create an infinite loop:

```
int i = 0;

while(i < 10) {

printf("%d ", i);

i++;
```

}

True / False

9. In a do-while loop, the condition is checked at the beginning of each iteration.

True / False

10. What will be the output of the following code?

```
int i = 1;
do {
printf("%d ", i);
i += 2;
} while(i <= 7);
```

a) 1 3 5 7

b) 1 3 5 7 9

c) 1 3 5

d) 2 4 6 8

Find the Error Questions

11. Find the error in the following code:

```
int i;
for(i = 0; i <= 10; i++) {
if(i == 5)
continue;
printf("%d ", i);
if(i == 8)
break;
```

a) Missing closing brace }

b) Wrong condition in for loop

c) continue statement is used incorrectly

d) break statement should come before printf

12. Identify the error in this nested loop:

```
int i, j;
for(i = 0; i < 3; i++) {
for(j = 0; j < 2; j++);
printf("i=%d, j=%d\n", i, j);
}
```

a) Variable j is not initialized properly

b) Semicolon after the inner for loop terminates it prematurely

c) printf statement has wrong format specifiers

d) Outer loop condition is incorrect