X

NPTEL National Programme on Technology Enhanced Learning | NPAT 2017



NPAT 2017

- NPAT Information
- **Q1. Multiple of at least two digits**
- Q2. Triangle inequality
- Q3. Does an integer appear k times in first n positions of an array
- Q4. Maximum of 3 given integers
- Q5. Largest sum of m consecutive elements
- Q6. Additive sequence of an array
- Q7. Printing 2n lines in reverse sequence
- Q8, Most frequent 3 integers

Practice Corner

# Q1. Multiple of at least two digits

Due on 2017-08-13, 18:00 IST

Compiler Information
Sum of two numbers
Solution for Sum of two numbers
Factorial
Solution for Factorial
Reverse Words
Store Credit
Solution for Reverse Words
Leap Year

Weightage: 10%

**Note:** In this question, you have to understand the code that is given and provide a testcase as explained in the problem statement.

The following function takes an integer n >= 10 as an argument. It should return 1 if n is a multiple of at least two of its digits, and it should return 0 otherwise.

For example, the function should return 1 if n = 22 or n = 24, and it should return 0 if n = 23 or n = 42.

Give a value of n for which the given function does NOT work.

```
int check(int n) {
  int m = n;
  int digit;
  int count = 0;
  while(m > 0) {
    digit = m % 10;
    m = m / 10;
    if(n % digit == 0) {
      count++;
    }
  }
  if(count >= 2) {
    return 1;
  } else {
    return 0;
  }
}
```

Open up the code submission box below and write your test case where you would normally enter your code.

Your input should be a positive integer.

Sample Test Cases

| | Input | Output |
|---|---|---|
| Test Case 1 | | `correct` |

| | | |
|---|---|---|
| Test Case 2 | | correct |

Due Date Exceeded. You scored 100.0/100.

Your last recorded submission was :

```c
#include <stdio.h>
#include <regex.h>

int myinput =
100
;

char regex_format[] = "[1-9][0-9]*0[0-9]*";

int main() {
  regex_t emma;
  regmatch_t matches[20];
  int status;
  char myinput_str[100];

  sprintf(myinput_str, "%d", myinput);
  status = regcomp(&emma, regex_format, REG_EXTENDED);
  status = regexec(&emma, myinput_str, 20, matches, 0);
  if(myinput >= 10 && !status)
    printf("correct\n");
  else
    printf("wrong\n");

  return 0;
}

```

End