

SET B

1

FIRST SEMESTER MCA(2020 SCHEME)
PRACTICAL EXAMINATION JUNE-JULY 2021
20MCA131 PROGRAMMING LAB

REG NO: ICE 20MCA2029
DATE: 02 JULY 2021
TIME: 11:00 - 4:00

1. Generate Fibonacci series of N terms.

ALGORITHM:

- step 1: start.
- step 2: Declare variables f_1 & f_2 and and initialise 0 and 1 respectively.
- step 3: Declare variable $f_3 = f_1 + f_2$.
- step 4: print f_1 & f_2 .
- step 5: Read the range and store in n . declare $i = 3$.
- step 6: ~~for~~ Repeat step 7 to step 12 until $i \leq n$.
- step 7: print f_3 .
- step 8: $f_1 = f_2$.
- step 9: $f_2 = f_3$.
- step 10: $f_3 = f_1 + f_2$.
- step 11: $i = i + 1$.
- step 12: stop.

PROGRAM:

```
n = int(input("Enter the number of terms needed"))
f1, f2 = 0, 1
f3 = f1 + f2
print("Fibonacci series of first", n, "terms is:")
print(f1)
print(f2)
for i in range(3, n+1):
    print(f3)
    f1 = f2
    f2 = f3
    f3 = f1 + f2
```

OUTPUT:

expected:

Enter the number of terms: 6
Fibonacci series of 6 terms:

0
1
1
2
3
5

tested:

Enter the number of terms: 7
Fibonacci series of 7 terms:

0
1
1
2
3
5
8

(2)

2. Create a class Time with private attribute hour, minute and second. Overload '+' operator to find sum of 2 time.

ALGORITHM:

- step 1: start.
- step 2: Declare a class Time.
- step 3: Define a method ~~constructor~~ with attributes h, m and s:
 initialise self.hr=h,
 self.min=m,
 self.sec=s.
- step 4: Define a method add with attribute other:
 Declare variable tempsec = self.sec + other.sec.
 Declare variable tempmin = tempsec / 60.
~~Declare~~ self.sec = integer of (tempsec % 60).
- step 5: self.min = self.min + other.min + tempmin.
 temp hr = self.min / 60.
 self.min = integer of (self.min % 60).
 self.hr = integer of (self.hr + other.hr + temp hr)
- step 6: return time(self.hr + other.hr + temp hr)
- step 7: return string(self.hr) + 'hr' + string of (self.min) + 'min' + string (self.sec) + 'sec'
- step 8: stop.
- Main:
- step 1: start
- step 2: Declare variables a, b, c, x, y, z.
- step 3: Read values of hour of t1 into a. as integer
- step 4: Read value of minute of t1 into b. as integer.
- step 5: Read value of second of t1 into c as integer.
- step 6: Read value of hour of t2 into x as integer.
- step 7: Read value of minute of t2 into y as integer.
- step 8: Read value of second of t2 into z as integer.
- step 9: call ~~method~~ ^{class} time with attributes a, b, c into it which returned to variable t1.
- step 10: call ~~method~~ ^{class} time with attributes x, y, z into it which returned to variable t2.
- step 11: print values t1 + t2.
- step 12: stop.

PROGRAM:

```

class time:
    def __init__(self, h, m, s):
        self.hr = h
        self.min = m
        self.sec = s
    def add(self, other):
        tempsec = self.sec + other.sec
        tempmin = tempsec / 60
        self.sec = int(tempsec % 60)
        self.min = self.min + other.min + tempmin
        temphr = self.min / 60
        self.min = int(self.min % 60)
        self.hr = int(self.hr + other.hr + temphr)
        return time(self.hr, self.min, self.sec)
    def __str__(self):
        return str(self.hr) + 'hr' + str(self.min) + 'min' + str(self.sec) + 'sec'

a = int(input("Enter hour of t1: "))
b = int(input("Enter minute of t1: "))
c = int(input("Enter second of t1: "))
x = int(input("Enter hour of t2: "))
y = int(input("Enter minute of t2: "))
z = int(input("Enter second of t2: "))
t1 = time(a, b, c)
t2 = time(x, y, z)
print(t1 + t2)

```

OUTPUT:Expected:

Enter hour of t1: 1.
 Enter minute of t1: 23
 Enter second of t1: 10
 Enter hour of t2: 3
 Enter minute of t2: 22
 Enter second of t2: 31
 4hr 45min 41sec.

Tested:

Enter hour of t1: 2
 Enter minute of t1: 30
 Enter second of t1: 30
 Enter hour of t2: 3
 Enter minute of t2: 30
 Enter second of t2: 30
 6hr 1min 0sec.