|  |  |
| --- | --- |
| **Ex. No. 4**  **Date:26.05.2021** | **CONDITIONAL CONSTRUCTS – LEVEL 2** |

**AIM:**

To write Python programs using branching and looping statements.

**PROGRAMS:**

**a) To check if a given number is Happy number or not**

**Description:**

A number is called happy if it leads to 1 after a sequence of steps wherein each step number is replaced by thesum of squares of its digit that is if we start with Happy Number and keep replacing it with digits square sum, we reach 1.

Example:

n = 19

19 is Happy Number,

1^2 + 9^2 = 82

8^2 + 2^2 = 68

6^2 + 8^2 = 100

1^2 + 0^2 + 0^2 = 1

As we reached to 1, 19 is a Happy Number.

n = 20

20 is not Happy Number

Sample Input

19

Sample Output

19 is a Happy Number

Sample Input

20

Sample Output

20 is not a Happy Number

**Program:**

‘’’Name: R.sridevi

Roll.No:20UIT021

Program Name: To check if a given number is Happy number or not.’’’

num=int(input())

def hn(num):

rem = sum = 0;

while(num > 0):

rem = num%10;

sum = sum + (rem\*rem);

num = num//10;

return sum;

result = num;

while(result != 1 and result != 4):

result = hn(result);

if(result == 1):

print(str(num) + " is a Happy Number");

elif(result == 4):

print(str(num) + " is not a Happy Number");

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 19 | 19 is a Happy Number |
| 2 | 20 | 20 is not a Happy Number |
| 3 | 68 | 68 is a Happy Number |
| 4 | 35 | 35 is not a Happy Number |
| **Total Test Cases** | | **4** |
| **Number of Test Cases Passed** | | **4** |

**b) To find Exponentiation (power of two given numbers) of the given number**

**Description:**

To find the exponentiation for a given positive number.

Sample Input

3

2

Sample Output

9

Sample Input

-2

1

Sample Output

The number is negative

**Program:**

‘’’Name: R.sridevi

Roll.No: 20UIT021

Program Name: To find exponentiation (power of two given numbers) of the given number.’’’

#variable declaration

num1=int(input())

num2=int(input())

if(num1>0 and num2>0):

exp=(num1\*\*num2)

print(exp)

else:

print("The number is negative")

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 3  2 | 9 |
| 2 | -2  1 | The number is negative |
| 3 | -4  -2 | The number is negative |
| 4 | 9  -3 | The number is negative |
| 5 | 4  7 | 16384 |
| 6 | 4  3 | 64 |
| **Total Test Cases** | | **6** |
| **Number of Test Cases Passed** | | **6** |

**c) To generate ‘n’ prime numbers**

**Description:**

To generate 'n' prime number between given range. Get the starting and ending limit.

Sample Input:

10

20

Sample Output:

11

13

17

19

Sample Input:

20

10

Sample Output:

The starting limit is greater than ending limit

Sample Input:

-3

10

Sample Output:

The number is negative

**Program:**

‘’’Name: R.sridevi

Roll.No: 20UIT021

Program Name: To generate ‘n’ prime numbers.’’’

#variable declaration

num1 = int(input())

num2 = int(input())

if num2>num1:

if num1>0 and num2>0:

for num in range(num1,num2+1):

if(num>1):

for i in range(2,num):

if(num%i==0):

break

else:

print(num)

else:

print('The number is negative')

else:

print('The starting limit is greater than ending limit')

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 10  20 | 11  13  17  19 |
| 2 | 20  10 | The starting limit is greater than ending limit |
| 3 | -3  10 | The number is negative |
| 4 | 1  10 | 2  3  5  7 |
| **Total Test Cases** | | **4** |
| **Number of Test Cases Passed** | | **4** |

**RESULT:**

Thus, the Python programs are executed successfully.