

Trần Tấn Phát

MSSV: 2274802010644

LAB6

Bài làm

BT1. Tự động kiểm tra và tạo thư mục nếu chưa tồn tại

```
GNU nano 6.2 BT5.py
import os

folder_name = "data_folder"

if not os.path.exists(folder_name):
    os.makedirs(folder_name)
    print(f"Đã tạo thư mục {folder_name}")
else:
    print(f"Thư mục đã tồn tại.")

ubuntu@ubuntu-2274802010644:~/iDragonCloud$ python3 BT5.py
Đã tạo thư mục data_folder
```

BT2: Tính tổng của tất cả các số chẵn từ 1 đến N

```
N = int(input("Nhập số N: "))
tong = sum(i for i in range(1, N+1) if i % 2 == 0)
print(f"Tổng các số chẵn từ 1 đến {N} là: {tong}")
█

ubuntu@ubuntu-2274802010644:~/iDragonCloud$ python3 BT5.py
Thư mục đã tồn tại.
Nhập số N: 10
Tổng các số chẵn từ 1 đến 10 là: 30
_
```

BT3: In bảng cửu chương

```
for i in range(1, 10):
    for j in range(1, 10):
        print(f"{i} x {j} = {i*j}", end="\t")
    print()
█
```

1 x 1 = 1	1 x 2 = 2	1 x 3 = 3	1 x 4 = 4	1 x 5 = 5	1 x 6 = 6	1 x 7 = 7	1 x 8 = 8	1 x 9
9								
2 x 1 = 2	2 x 2 = 4	2 x 3 = 6	2 x 4 = 8	2 x 5 = 10	2 x 6 = 12	2 x 7 = 14	2 x 8 = 16	2 x 9
18								
3 x 1 = 3	3 x 2 = 6	3 x 3 = 9	3 x 4 = 12	3 x 5 = 15	3 x 6 = 18	3 x 7 = 21	3 x 8 = 24	3 x 9
27								
4 x 1 = 4	4 x 2 = 8	4 x 3 = 12	4 x 4 = 16	4 x 5 = 20	4 x 6 = 24	4 x 7 = 28	4 x 8 = 32	4 x 9
36								
5 x 1 = 5	5 x 2 = 10	5 x 3 = 15	5 x 4 = 20	5 x 5 = 25	5 x 6 = 30	5 x 7 = 35	5 x 8 = 40	5 x 9
45								
6 x 1 = 6	6 x 2 = 12	6 x 3 = 18	6 x 4 = 24	6 x 5 = 30	6 x 6 = 36	6 x 7 = 42	6 x 8 = 48	6 x 9
54								
7 x 1 = 7	7 x 2 = 14	7 x 3 = 21	7 x 4 = 28	7 x 5 = 35	7 x 6 = 42	7 x 7 = 49	7 x 8 = 56	7 x 9
63								
8 x 1 = 8	8 x 2 = 16	8 x 3 = 24	8 x 4 = 32	8 x 5 = 40	8 x 6 = 48	8 x 7 = 56	8 x 8 = 64	8 x 9
72								

BT4: Tính giai thừa của một số N (Đệ quy)

```
def giai_thua(n):
    if n == 0 or n == 1:
        return 1
    return n * giai_thua(n - 1)
n = int(input("Nhap so N"))
print (f" Giai thua cua {n} la : {giai_thua(n)}")
```

```
Nhap so N10
Giai thua cua 10 la : 3628800
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

BT5: In số Fibonacci lên đến N (Dùng vòng lặp và mảng)

```
N = int(input("Nhap N: "))
fib = [0, 1]

for i in range(2, N):
    fib.append(fib[i-1] + fib[i-2])

print("Day Fibonacci:", fib[:N])
```

```
Nhap N: 10
Day Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

BT6: Tìm số lớn nhất trong một danh sách số nhập từ bàn phím

```
numbers = list(map(int, input("Nhap cac so , cach nhau bang khoang trang:").split()))
print (f" So lon nhat la: {max(numbers)}")
```

```
Nhập các số , cách nhau bằng khoảng trắng: 2 4 2 1 4 5 6 19 28 20
Số lớn nhất là: 28
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

BT7: Kiểm tra số hoàn hảo

```
n = int(input("Nhập số : "))
tong = sum(i for i in range(1, n) if n % i == 0)
if tong == n:
    print(f"{n} là số hoàn hảo")
else:
    print(f"{n} không phải số hoàn hảo")
```

```

Nhập số : 120
120 không phải số hoàn hảo
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

BT8: Kiểm tra số Armstrong

```
n = int(input("Nhập số: "))
s = sum(int(d)**3 for d in str(n))
if s == n:
    print(f"{n} là số Armstrong")
else:
    print(f"{n} không phải số Armstrong")
```

```

Nhập số: 153
153 là số Armstrong
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

BT9: Kiểm tra dãy số có phải là cấp số cộng không

```
nums = list(map(int, input("Nhập dãy số , cách nhau bằng khoảng trắng:").split()))
hieu = nums[1] - nums[0]
is_AP = all(nums[i+1] - nums[i] == hieu for i in range(len(nums)-1))
print(" là cấp số cộng" if is_AP else " Không phải cấp số cộng")
```

```

Nhập dãy số , cách nhau bằng khoảng trắng: 1 2 5 3 12 24 52 23 11 54
Không phải cấp số cộng
```

BT10: Kiểm tra số Palindrome nhưng chỉ dùng vòng lặp (Không dùng Rev)

```

n = input("Nhap so: ")
is_palindrome = True

for i in range(len(n)//2):
    if n[i] != n[-i-1]:
        is_palindrome = False
        break
print(f"{n} la Palindrome" if is_palindrome else f"{n} : khong phai Palindrome")

```

```

Nhap so: 123
123 : khong phai Palindrome

```

BT11: Viết hàm kiểm tra số nguyên tố

```

GNU nano 6.2                                     BT5.py *
def la_so_nguyen_to(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
n = int(input("Nhap 1 so de kiểm tra so nguyên to:"))
if la_so_nguyen_to(n):
    print(f"{n} la so nguyên to")
else:
    print(f"{n} khong phai so nguyên to")

```

```

12 : khong phai Palindrome
Nhap 1 so de kiểm tra so nguyên to: 90
90 khong phai so nguyên to

```

Bài 12: Viết hàm đảo ngược một chuỗi

```

def dao_nguoc_chuoi(s):
    dao = ""
    for ch in s:
        dao = ch + dao
    return dao

s = input("Nhap chuỗi: ")
print("Chuỗi đảo ngược:", dao_nguoc_chuoi(s))

```

```

Nhap chuỗi: 12 2 5 61 123 54 122
Chuỗi đảo ngược: 221 45 321 16 5 2 21

```