UCS1403 Design and Analysis of Algorithms
AY: 2021-22

Name: *Krithika Swaminathan*
Reg. no.: *205001057*

DAA Lab – Assignment 9

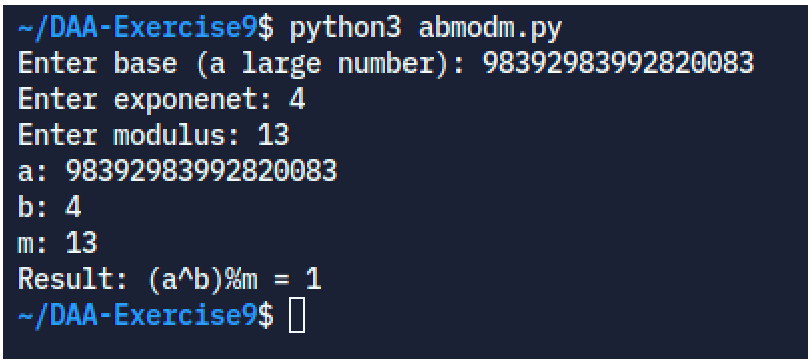
Fast Exponentiation and Primality Check

1. To Compute a^b mod N.

Code:

```
# Program to find (a^b) mod m with fast exponentiation

def aModM(s,mod):
	num = 0
	for i in range(len(s)):
		num = (num*10 + int(s[i]))
		num = num % m
	return num

def ApowBmodM(a,b,m):
	ans = aModM(a,m)
	mul = ans
	for i in range(1,b):
		ans = (ans*mul) % m
	return ans

a = input("Enter base (a large number): ")
b = int(input("Enter exponenet: "))
m = int(input("Enter modulus: "))

print("a:",a)
print("b:",b)
print("m:",m)
print ("Result: (a^b)%m =",ApowBmodM(a,b,m))
```

Output:



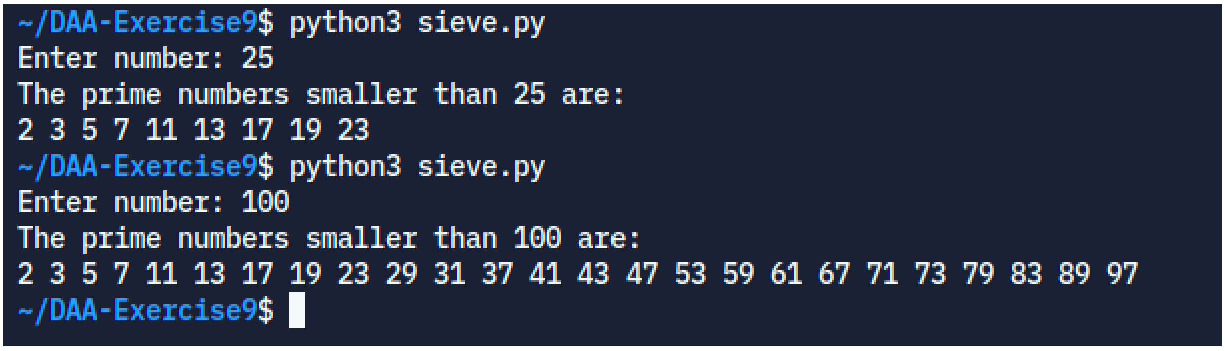
```
~/DAA-Exercise9$ python3 abmodm.py
Enter base (a large number): 98392983992820083
Enter exponenet: 4
Enter modulus: 13
a: 98392983992820083
b: 4
m: 13
Result: (a^b)%m = 1
~/DAA-Exercise9$ 
```

2. To Implement Sieve of Eratosthenes Algorithm.

Code:

```
# Program to implement the Sieve of Eratosthenes algorithm for primality
check on numbers in a given range

def SieveOfEratosthenes(n):
      prime = [True for i in range(n+1)]
      p = 2
      while (p * p <= n):
            if (prime[p] == True):
                  for i in range(p * p, n+1, p):
                        prime[i] = False
            p += 1

      # printing the prime numbers
      for p in range(2, n+1):
            if prime[p]:
                  print(p,end=' ')
      print()

n = int(input("Enter number: "))
print("The prime numbers smaller than",n,"are: ")
SieveOfEratosthenes(n)
```

Output:



```
~/DAA-Exercise9$ python3 sieve.py
Enter number: 25
The prime numbers smaller than 25 are:
2 3 5 7 11 13 17 19 23
~/DAA-Exercise9$ python3 sieve.py
Enter number: 100
The prime numbers smaller than 100 are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
~/DAA-Exercise9$
```