

Câu 1 Một số gọi là Q-prime khi nó có đúng 4 ước số nguyên dương. Hãy viết chương trình in ra các số Q-Prime nhỏ hơn hoặc bằng một số N cho trước nhập từ bàn phím.

Câu 2. Viết chương trình tìm các số nguyên tố có N chữ số với N nhập từ bàn phím và $2 \leq N \leq 10$.

Câu 3. Cho một số nguyên dương N, gọi:

- k là số ước nguyên tố của N;
- q là tổng của các ước nguyên tố của N;
- p là tổng của các ước số của N;
- s là số ước của N;

Hãy viết chương trình tính giá trị của: $N+p+s-q-k$ với N cho trước nhập từ bàn phím.

Ví dụ: $N=24$, có các ước là $\{1,2,3,4,6,8,12, 24\}$ do đó:

$p=1+2+3+4+6+8+12+24=60$ và $s=8$

trong đó có 2 ước nguyên tố là $\{2,3\}$ do đó:

$q=2+3=5$ và $k=2$

Và từ đó: $N+p+s-q-k = 24+60+8-5-2=85$;

Câu 4. Viết chương trình đếm số số nguyên tố nằm trong khoảng $[A,B]$ với A, B nhập vào từ bàn phím.

Câu 5. Viết chương trình tính tổng của các số nguyên tố nằm trong khoảng $[A, B]$ với A, B nhập vào từ bàn phím.

Câu 6. Hai số tạo thành một cặp số thân thiết khi chúng tuân theo quy luật: Số này bằng tổng tất cả các ước của số kia (trừ chính số đó) và ngược lại. Viết chương trình tìm hai số dạng này nhỏ hơn N (với N nhập vào từ bàn phím).

Câu 7. Một số emirp là một số nguyên tố mà khi đảo ngược vị trí các chữ số của nó, ta cũng được một số nguyên tố. Viết chương trình liệt kê các số emirp nhỏ hơn N với N nhập vào từ bàn phím.

Câu 8. Một số gọi là số T-prime nếu có đúng 3 ước nguyên dương. Viết chương trình tìm các số T-prime nhỏ hơn hoặc bằng N với N cho trước nhập từ bàn phím.

Câu 9. Viết chương trình đếm số số nguyên tố nhỏ hơn hoặc bằng N với N được nhập vào từ bàn phím.

Câu 10. Viết chương trình đếm số ước và số ước nguyên tố của một số N nhập vào từ bàn phím.

Câu 11. Viết chương trình tính tổng của các số nguyên tố nhỏ hơn hoặc bằng N với N được nhập từ bàn phím.

Câu 12. Viết chương trình tìm bốn số nguyên tố liên tiếp, sao cho tổng của chúng là số nguyên tố nhỏ hơn hoặc bằng N (với N được nhập vào từ bàn phím).

Câu 13. Viết chương trình tìm hai số nguyên tố nhỏ hơn hoặc bằng N với N nhập vào từ bàn phím, sao cho tổng và hiệu của chúng đều là số nguyên tố.

Câu 14. Viết chương trình tìm số nguyên tố có ba chữ số, biết rằng nếu viết số đó theo thứ tự ngược lại thì ta được một số là lập phương của một số tự nhiên.

Câu 15. Viết chương trình Hai số nguyên tố sinh đôi là hai số nguyên tố hơn kém nhau 2 đơn vị. Tìm hai số nguyên tố sinh đôi nhỏ hơn hoặc bằng N, với N được nhập vào từ bàn phím.

Câu 16. Viết chương trình tìm các số nguyên tố từ một mảng sinh ngẫu nhiên có kích thước N, với N nhập vào từ bàn phím.

Câu 17. Viết chương trình tìm số nguyên dương x nhỏ nhất sao cho giá trị của biểu thức $Ax^2 + Bx + C$ là một số nguyên tố với A, B, C là các số nguyên nhập vào từ bàn phím.

Câu 18. Áp dụng thuật toán đã được học để viết chương trình tính tổng của hai số nguyên lớn, hiển thị dưới dạng mảng và dạng số nguyên.

Câu 19. Viết chương trình in ra các số nguyên dương nằm trong khoảng $[m, l]$ sao cho giá trị của biểu thức $Ax^2 + Bx + C$ là một số nguyên tố. Với A, B, C, m, l là các số nguyên nhập từ bàn phím ($m < l$).

Câu 20. Viết chương trình in ra các cặp số (A, B) nằm trong khoảng (M, N) sao cho ước số chung lớn nhất của A và B có giá trị là một số D cho trước. Với M, N, D nhập vào từ bàn phím. ($0 < M, N, D < 1000$).

Câu 21. Một số gọi là siêu số nguyên tố nếu số lượng các số nguyên tố từ 1 đến X (ngoại trừ X) là một số nguyên tố. Hãy viết chương trình đếm số lượng các siêu số nguyên tố này trong khoảng $[A, B]$ cho trước nhập từ bàn phím.

Câu 22. Với một số nguyên dương N thỏa mãn $0 < N < 10000$, đặt:

$F(N) = N$ nếu N là một số nguyên tố

$F(N) = 0$ nếu N là hợp số

Cho L và R nhập vào từ bàn phím, với mọi cặp i, j trong khoảng $[L, R]$ hãy viết chương trình in ra màn hình giá trị của tổng $F(i) * F(j)$ với $j > i$.

Câu 23. Viết chương trình in ra màn hình YES trong trường hợp tổng của các số nguyên tố trong khoảng $[A, B]$ là cũng là một số nguyên tố và NO nếu ngược lại. Với A, B là hai số được nhập vào từ bàn phím.

Câu 24. Đặt $S1, S2$ là các mảng chứa giá trị bình phương của các số nguyên. Hãy viết chương trình in ra số lượng tất cả các số nguyên tố nằm trong khoảng $[a, b]$ sao cho số này cũng là tổng của hai số x và y với x thuộc $S1$ và y thuộc $S2$. Trong đó, a, b là các số được nhập từ bàn phím
Ví dụ: với $a=10, b=15$, in ra giá trị là 1 vì trong khoảng $[10, 15]$ chỉ có 2 số nguyên tố 11 và 13, nhưng chỉ có $13 = 2^2 + 3^2 = 4 + 9$.

Câu 25. Cho 2 số M và N thỏa mãn điều kiện: $1 \leq N \leq 10000$; $2 \leq M \leq 100$; Hãy viết chương trình xác định xem số N có thể được phân tích thành tổng của M số nguyên tố hay không? Nếu có thì in ra các số đó.

Ví dụ: $N=10$ và $M=3$, thì $10=2+3+5$ do đó kết quả trả về là thỏa mãn và in ra 3 số 2, 3, 5.

Câu 26. Một số được gọi là số mạnh mẽ khi nó đồng thời vừa chia hết cho số nguyên tố và chia hết cho bình phương của số nguyên tố đó. Tìm số mạnh mẽ nhỏ hơn số N cho trước ($N < 10000$).

PHẦN II. PHẦN NÂNG CAO (6 điểm)

Câu 27. Viết chương trình in ra các cặp số (a, b) thỏa mãn điều kiện $0 < a, b < 1000$, sao cho ước chung lớn nhất của 2 số đó là một số nguyên tố.

Câu 28. Viết chương trình tìm các số Carmichael (là các số giả nguyên tố n thỏa mãn điều kiện là hợp số và thỏa mãn $b^{n-1} \equiv 1 \pmod{n}$ với mọi số nguyên dương b nguyên tố cùng nhau với n) nhỏ hơn một số N cho trước nhập vào từ bàn phím (với điều kiện $0 \leq N \leq 10000$).

Câu 29. Viết chương trình đếm số các số Carmichael (là các số giả nguyên tố n thoả mãn điều kiện là hợp số và thoả mãn $b^{n-1} \equiv 1 \pmod{n}$ với mọi số nguyên dương b nguyên tố cùng nhau với n) nhỏ hơn một số N cho trước nhập vào từ bàn phím (với điều kiện $0 \leq N \leq 10000$).

Câu 30. Viết chương trình tính tổng của các số Carmichael (là các số giả nguyên tố n thoả mãn điều kiện là hợp số và thoả mãn $b^{n-1} \equiv 1 \pmod{n}$ với mọi số nguyên dương b nguyên tố cùng nhau với n) nhỏ hơn một số N cho trước nhập vào từ bàn phím (với điều kiện $0 \leq N \leq 10000$).

Câu 31. Áp dụng theo các **thuật toán đã được học** trong phần lí thuyết em hãy cài đặt chương trình:

- Tìm số nguyên tố k gần nhất với phần số của mã số sinh viên của mình (trong trường hợp khoảng cách bằng nhau thì lấy số nhỏ hơn).
- Từ số k tìm được tính $a^k \bmod n$ với $a = \text{SBD}$, $n = 123456$.

Câu 32. Áp dụng các **thuật toán đã được học** em hãy cài đặt chương trình giải bài toán mô phỏng cách mã và giải mã của hệ mật RSA như sau:

- Tìm số nguyên số p, q (trong đó $100 < p, q < 500$)
- Tính $n = p \cdot q$; $\phi(n) = (p - 1)(q - 1)$
- Chọn e là số nguyên tố cùng nhau với $\phi(n)$ ($\gcd(e, \phi(n)) = 1$) và tính $d = e^{-1} \bmod \phi(n)$
- Tính bản mã c của thông điệp m , với $m = \text{SBD} + 123$, $c = m^e \bmod n$
- Giải mã thông điệp, tính $m = c^d \bmod n$

Câu 33. Áp dụng thuật toán Euclide mở rộng đã được học tìm đa thức nghịch đảo trên trường hữu hạn $\text{GF}(2^3)$ với đa thức bất khả quy $g(x) = x^3 + x + 1$.

Input: $a(x), g(x)$

Output: $a^{-1}(x)$ thoả mãn $a^{-1}(x) \cdot a(x) \bmod g(x) = 1$

Các phép cộng và nhân trong $\text{GF}(2^3)$ được định nghĩa như sau:

- **Phép cộng** = phép XOR = mod 2
 - o $a(x) = x^2 + 1$; $b(x) = x^2 + x + 1 \Rightarrow a(x) + b(x) = \underline{x^2} + \underline{1} + \underline{x^2} + x + \underline{1} = x$
- **Phép nhân:** nhân thông thường sau đó KQ rút gọn cho đa thức $g(x)$
 - o $a(x) = x^2 + 1$; $b(x) = x^2 + x + 1$

$$\Rightarrow a(x) \cdot b(x) = x^4 + x^3 + \underline{x^2} + \underline{x^2} + x + 1 = x^4 + x^3 + x + 1 \bmod (x^3 + x + 1) = x^2 + x$$

VD: Input: $a(x) = x^2 + x + 1$; $g(x) = x^3 + x + 1$

\Rightarrow Output: $a^{-1}(x) = x^2$

q	r	x	y	a	b	x ₂	x ₁	y ₂	y ₂
-	-	-	-	$x^3 + x + 1$	$x^2 + x + 1$	1	0	0	1
$x + 1$	x	1	$x + 1$	$x^2 + x + 1$	x	0	1	1	$x + 1$

$x + 1$	1	$x + 1$	x^2	x	1	1	$x + 1$	$x + 1$	x^2
x	0	$x^2 + x + 1$	1	1	0	$x + 1$	$x^2 + x + 1$	x^2	1

Câu 34. Cài đặt thuật toán kiểm tra số nguyên tố Fermat. Trong trường hợp nào thì số nào thì thuật toán cho kết quả kiểm tra sai.

Câu 35. Cài đặt thuật toán kiểm tra số nguyên tố Miller-Rabin in ra kết luận về 1 số nguyên dương N nhập vào từ bàn phím với xác suất kết luận tương ứng sau thuật toán.

Câu 36. Lập trình tìm kiếm xâu S1 trong xâu S2 theo thuật toán Boyer-Moore, in giá trị của bảng. Trong trường hợp nào thì thuật toán Boyer-Moore được xem là cải tiến hơn thuật toán tìm kiếm vét cạn.

Câu 37. Lập trình tìm kiếm xâu S1 trong xâu S2 theo thuật toán Knutt-Morris-Patt. Trong trường hợp nào thì thuật toán Boyer-Moore được xem là cải tiến hơn thuật toán tìm kiếm vét cạn?

Câu 38. Tìm nghịch đảo của một số a trong trường F_p với a và p được nhập từ bàn phím.

Câu 39. Cho mảng A nhập từ bàn phím gồm các số nguyên dương. Hãy viết chương trình tìm các cặp số (i,j) trong mảng A sao cho ước chung lớn nhất của chúng là một số nguyên tố.

Câu 40. Cho mảng A nhập từ bàn phím gồm các số nguyên dương. Hãy viết chương trình đếm các cặp số (i,j) trong mảng A sao cho ước chung lớn nhất của chúng là một số nguyên tố.

Câu 41. Cho các số nguyên dương a,k,n, nhập từ bàn phím ($0 < a, k < n < 1000$), Viết chương trình xác định xem $a^k \bmod n$ có phải là một số nguyên tố hay không (sử dụng thuật toán bình phương và nhân có lặp)?

Câu 42. Viết chương trình sinh ra 2 số nguyên tố $0 < p, q < 1000$ và kiểm tra với với số $0 < a < 100$ thì những số nào thoả mãn: $a^p \bmod q$ là số nguyên tố.

Câu 43. Cho N nhập vào từ bàn phím ($0 < N < 1000$), hãy viết chương trình tìm tất cả các số nguyên a < N sao cho $a^p \bmod N$ là số nguyên tố.

Câu 44. Cho mảng A gồm các số nguyên thuộc F_p nhập vào từ bàn phím, hãy viết chương trình in ra mảng B có các phần tử là nghịch đảo của các phần tử tương ứng trong A.

Câu 45. Viết chương trình sinh một mảng số nguyên tố A gồm N phần tử (N nhập từ bàn phím) sử dụng kiểm tra Miller-Rabin. In ra mảng và tính khoảng cách nhỏ nhất giữa 2 số bất kỳ trong mảng.