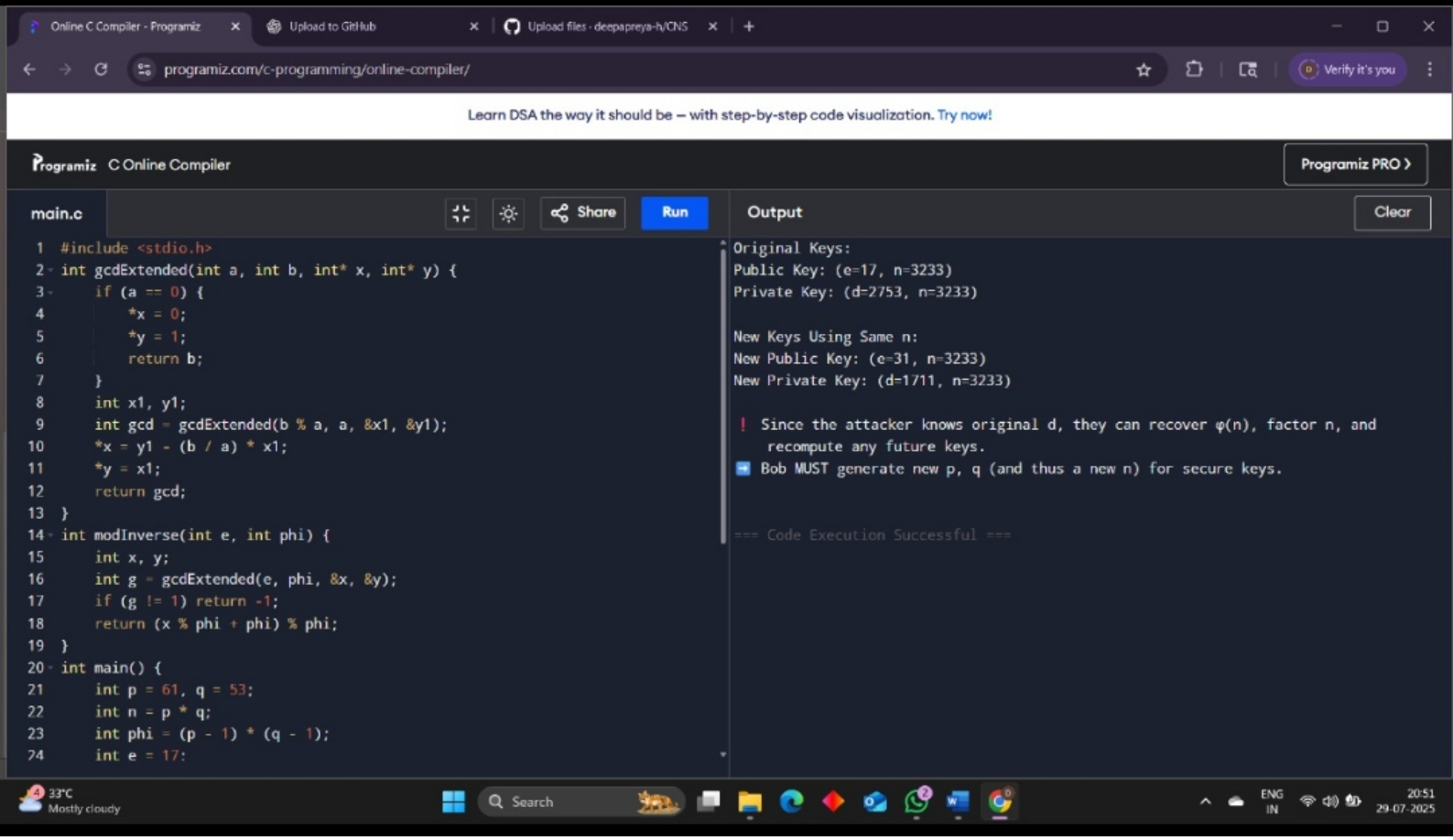
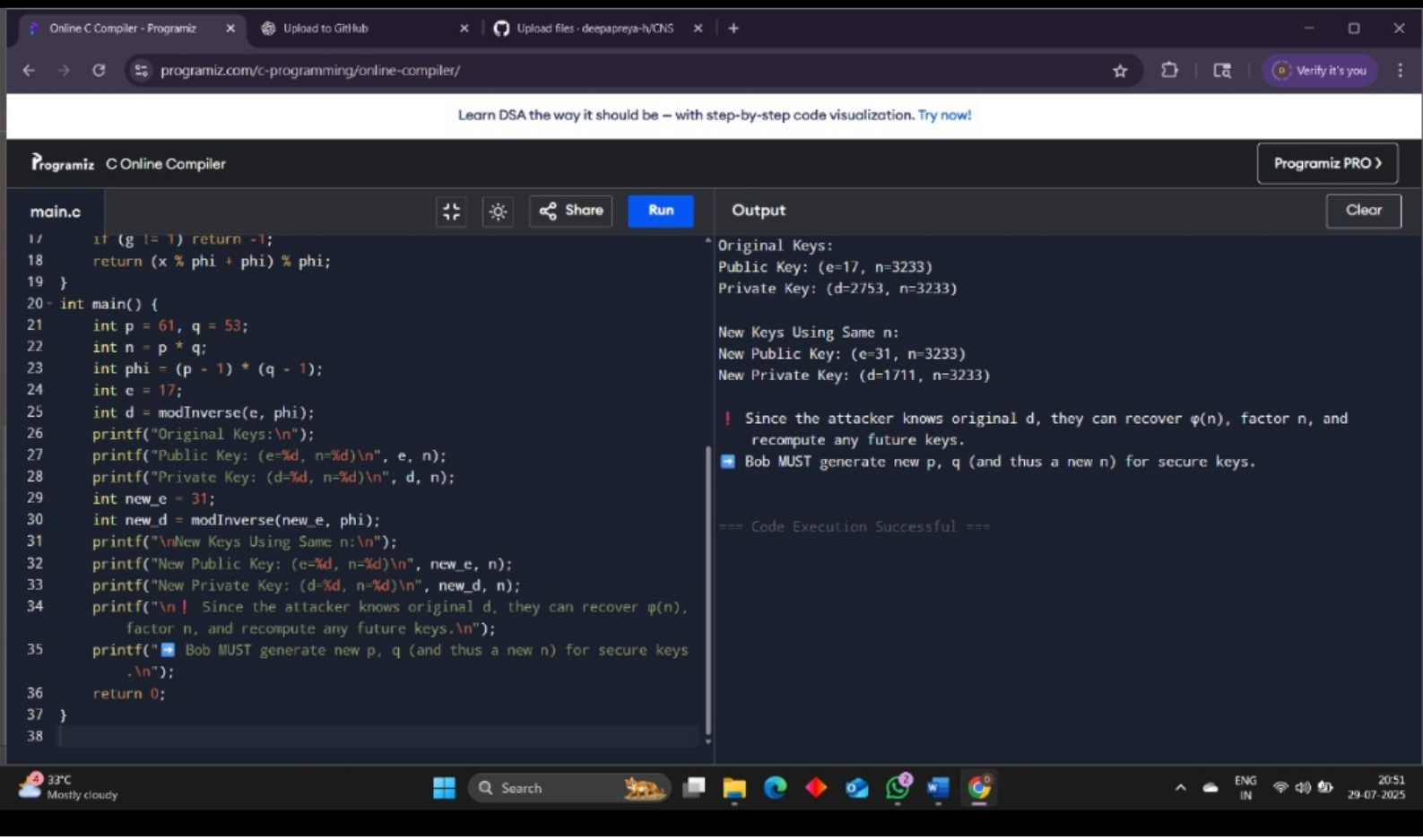

Online C Compiler - Programiz
Upload to GitHub
Upload files · deepapreya-h/CNS
programiz.com/c-programming/online-compiler/
Verify it's you
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
int gcdExtended(int a, int b, int* x, int* y) {
if (a == 0) {
*x = 0;
*y = 1;
return b;
}
int x1, y1;
int gcd = gcdExtended(b % a, a, &x1, &y1);
*x = y1 - (b / a) * x1;
*y = x1;
return gcd;
}
int modInverse(int e, int phi) {
int x, y;
int g = gcdExtended(e, phi, &x, &y);
if (g != 1) return -1;
return (x % phi + phi) % phi;
}
int main() {
int p = 61, q = 53;
int n = p * q;
int phi = (p - 1) * (q - 1);
int e = 17;
Original Keys:
Public Key: (e=17, n=3233)
Private Key: (d=2753, n=3233)
New Keys Using Same n:
New Public Key: (e=31, n=3233)
New Private Key: (d=1711, n=3233)
! Since the attacker knows original d, they can recover φ(n), factor n, and recompute any future keys.
➡ Bob MUST generate new p, q (and thus a new n) for secure keys.
=== Code Execution Successful ===
33°C
Mostly cloudy
Search
ENG
IN
20:51
29-07-2025


Online C Compiler - Programiz
Upload to GitHub
Upload files · deepapreya-h/CNS
programiz.com/c-programming/online-compiler/
Verify it's you
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
if (g != 1) return -1;
return (x % phi + phi) % phi;
}
int main() {
int p = 61, q = 53;
int n = p * q;
int phi = (p - 1) * (q - 1);
int e = 17;
int d = modInverse(e, phi);
printf("Original Keys:\n");
printf("Public Key: (e=%d, n=%d)\n", e, n);
printf("Private Key: (d=%d, n=%d)\n", d, n);
int new_e = 31;
int new_d = modInverse(new_e, phi);
printf("\nNew Keys Using Same n:\n");
printf("New Public Key: (e=%d, n=%d)\n", new_e, n);
printf("New Private Key: (d=%d, n=%d)\n", new_d, n);
printf("\n ! Since the attacker knows original d, they can recover φ(n), factor n, and recompute any future keys.\n");
printf("➡ Bob MUST generate new p, q (and thus a new n) for secure keys.\n");
return 0;
}
Original Keys:
Public Key: (e=17, n=3233)
Private Key: (d=2753, n=3233)
New Keys Using Same n:
New Public Key: (e=31, n=3233)
New Private Key: (d=1711, n=3233)
! Since the attacker knows original d, they can recover φ(n), factor n, and recompute any future keys.
➡ Bob MUST generate new p, q (and thus a new n) for secure keys.
=== Code Execution Successful ===
33°C
Mostly cloudy
Search
ENG
IN
20:51
29-07-2025