**(1) Multiples of 3 or 5**

#include<stdio.h>

int main(){

int sum=0,i;

for(i=1;i<1000;i++){

if(i%3==0||i%5==0){

sum=sum+i;

}

}

printf("%d",sum);

}

**Output:** 233168

**(2) Even Fibonnacci sequence**

#include<stdio.h>

int main(){

int sum=0,a=1,b=1,c=a+b;

while(c<4000000){

sum+=c;

a=b+c;

b=c+a;

c=a+b;

}

printf("%d",sum);

}

**Output:** 4613732

**(3) Largest Prime Factor**

#include<stdio.h>

#include<math.h>

int prime(long int n){

int i,max;

while(n%2==0){

max=2;

n=n/2;

}

for(i=3;i<=sqrt(n);i+=2){

while(n%i==0){

max=i;

n=n/i;

}

}

if(n>2){

max=n;

}

printf("%d",max);

}

int main(){

prime(600851475143);

}

**Output:** 6857

**(4) Palindromic Product**

#include <stdio.h>

int test\_palindromic(int n);

int main(void)

{

int i, j, max = 0;

for (i = 100; i <= 999; i++) {

for (j = 100; j <= 999; j++) {

int p = i\*j;

if (test\_palindromic(p) && p > max) {

max = p;

}

}

}

printf("%u\n", max);

return 0;

}

int test\_palindromic(int n)

{

unsigned int reversed\_num = 0, t = n;

while (t) {

reversed\_num = 10\*reversed\_num + (t % 10);

t /= 10;

}

return reversed\_num == n;

}

**Output:** 906609

**(5) Smallest Multiple**

#include<stdio.h>

int gcd (int a, int b)

{

while (b != 0)

{

a %= b;

a ^= b;

b ^= a;

a ^= b;

}

return a;

}

int lcm(int a, int b)

{

return a / gcd(a, b) \* b;

}

int main()

{

int res = 1;

for (int i = 2; i <= 20; i++)

{

res = lcm(res, i);

}

printf("%d\n", res);

return 0;

}

**Output:** 232792560

**(6) Sum square difference**

#include<stdio.h>

int main(){

int i,sum1=0,sum2,total=0,diff;

for(i=1;i<=100;i++){

sum1=sum1+(i\*i);

}

for(i=1;i<=100;i++){

total=total+i;

}

sum2=(total)\*(total);

diff=sum2-sum1;

printf("%d",diff);

}

**Output:** 25164150

**(7) 10001st prime**

#include<stdio.h>

#include<math.h>

int main(){

int n=10001,i,c=0,count,num=2,latest;

while(c!=n){

int count=0;

for(i=2;i<=sqrt(num);i++){

if(num%i==0){

count++;

break;

}

}

if(count==0){

c++;

latest=num;

}

num=num+1;

}

printf("%d",latest);

}

**Output:** 104743

**(9) Special Pythagorean triplet**

#include <stdio.h>

int main()

{

int n=1000;

pythagoreanTriplet(n);

return 0;

}

void pythagoreanTriplet(int n){

for(int i=1;i<=n/3;i++){

for(int j=i+1;j<=n/2;j++){

int k=n-i-j;

if((i\*i)+(j\*j)==(k\*k)){

int l=i\*j\*k;

printf("%d",l);

return;

}

}

}

printf("No Triplets");

}

**Output:** 31875000