Problems

1. **If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.** Ans: 233168

#include <stdio.h>

int main()

{

int sum=0,i;

for (int i=0; i<1000; i++)

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

sum=sum+i;

printf("%d", sum);

return 0;

}

1. **Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms**. Ans: 4613732

#include<stdio.h>

void main(){

int fibonacci =0, a1=0, a2=1, sum = 0;

printf("The fibonacci sequence is: ");

while(fibonacci<=4000000){

fibonacci = a1+a2;

if(fibonacci%2==0){

sum = sum+fibonacci;

}

a1=a2;

a2=fibonacci;

printf("%d\t",fibonacci);

}

printf("The sum of the series : %d", sum);

}

1. **The prime factors of 13195 are 5, 7, 13 and 29. What is the largest prime factor of the number 600851475143?** Ans: 6857

#include <stdio.h>

int main() {

long int n;

n=608851475143;

long int div=2, ans= 0, maxFact;

while(n!=0) {

if(n %div !=0)

div=div + 1;

else {

maxFact=n;

n=n/ div;

if(n = 1) {

printf("%d is the largest prime factor !",maxFact);

ans =1;

break;

}

}

}

return 0;

}

1. A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99. Find the largest palindrome made from the product of two 3-digit numbers. Ans: 906609

n = 0

for a in range(999, 100, -1):

for b in range(a, 100, -1):

x = a \* b

if x > n:

s = str(a \* b)

if s == s[::-1]:

n = a \* b

print(n)

1. 2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder. What is the smallest positive number that is evenly divisible by all of the numbers from 1 to 20? Ans: 232792560

def gcd(x,y): return y and gcd(y, x % y) or x

def lcm(x,y): return x \* y / gcd(x,y)

n = 1

for i in range(1, 21):

n = lcm(n, i)

print(n)

1. The sum of the squares of the first ten natural numbers is, 1^2+2^2+3^2 … + 10^2 = 385 The square of the sum of the first ten natural numbers is, (1 + 2 + 3 + … + 10) ^2 = 3025 Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is 3025 - 385 = 2640. Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum. Ans: 25164150

#include <stdio.h>

int main(void)

{

unsigned sum1=0, sum2 =0, i;

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

sum1 += i\*i;

sum2 += i;

}

printf("%u\n", sum2\*sum2- sum1);

return 0;

}

1. By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th prime is 13. What is the 10,001st prime number? Ans: 104743

#include <stdio.h>

int main(void){

long count, currentNumber, primeCount;

primeCount = 1;

for (currentNumber = 3; primeCount != 10001; currentNumber = currentNumber + 2)

{

for (count = 3; count < currentNumber; count = count + 2)

{

if (currentNumber % count == 0)

break;

}

if (count == currentNumber)

primeCount++;

}

printf("10001st Prime number: %li", count);

return 0;

}

1. A Pythagorean triplet is a set of three natural numbers, a < b < c, for which, a2 + b2 = c2 For example, 32 + 42 = 9 + 16 = 25 = 52. There exists exactly one Pythagorean triplet for which a + b + c = 1000. Find the product abc Ans: 31875000

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a,b,c,flag = 1;

int res = 0;

a = 1; b = 2; c = 3;

while (c<997 && flag ==1)

{b = 2;

while (b<c && flag ==1)

{ a = 1;

while (a<b && flag ==1)

{

if (((a\*a) + (b\*b))== (c\*c))

{

if ( (a + b + c) == 1000)

{

res= (a\*b) \*c;

printf("%d\*%d + %d\*%d = %d%d ,,,,,,,,,,,",a,a, b, b,c, c);

flag = 0;

}}

a++;

}

b++;

}

c++;

}

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

return 0;

}

1. The sum of the primes below 10 is 2 + 3 + 5 + 7 = 17. Find the sum of all the primes below two million. Ans: 142913828922

#include <stdio.h>

#include <math.h>

#define BELOW 2000000

int isaprime (int num);

int main (void) {

int i;

long sum = 0;

for (i = 2; i < BELOW; i++) {

if (isaprime(i) == 1) {

sum = sum + i;

}

}

printf("sum: %ld\n", sum);

return 0;

}

int isaprime (int num) {

int i;

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

if (num % i == 0) {

return 0;

}

else {

;

}

}

return 1;

}

1. In the 20×20 grid below, four numbers along a diagonal line have been marked in red.

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08 49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00 81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65 52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91 22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80 24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50 32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70 67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21 24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72 21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95 78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92 16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57 86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58 19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40 04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66 88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69 04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36 20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16 20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54 01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

The product of these numbers is 26 × 63 × 78 × 14 = 1788696. What is the greatest product of four adjacent numbers in the same direction (up, down, left, right, or diagonally) in the 20×20 grid? Ans: 70600674

#include <stdio.h>

#define N 20

static \_\_inline unsigned max(unsigned a, unsigned b);

int main(void)

{

unsigned grid[20][20] = {

{8,2,22,97,38,15,0,40,0,75,4,5,7,78,52,12,50,77,91,8},

{49,49,99,40,17,81,18,57,60,87,17,40,98,43,69,48,4,56,62,0},

{81,49,31,73,55,79,14,29,93,71,40,67,53,88,30,3,49,13,36,65},

{52,70,95,23,4,60,11,42,69,24,68,56,1,32,56,71,37,2,36,91},

{22,31,16,71,51,67,63,89,41,92,36,54,22,40,40,28,66,33,13,80},

{24,47,32,60,99,3,45,2,44,75,33,53,78,36,84,20,35,17,12,50},

{32,98,81,28,64,23,67,10,26,38,40,67,59,54,70,66,18,38,64,70},

{67,26,20,68,2,62,12,20,95,63,94,39,63,8,40,91,66,49,94,21},

{24,55,58,5,66,73,99,26,97,17,78,78,96,83,14,88,34,89,63,72},

{21,36,23,9,75,0,76,44,20,45,35,14,0,61,33,97,34,31,33,95},

{78,17,53,28,22,75,31,67,15,94,3,80,4,62,16,14,9,53,56,92},

{16,39,5,42,96,35,31,47,55,58,88,24,0,17,54,24,36,29,85,57},

{86,56,0,48,35,71,89,7,5,44,44,37,44,60,21,58,51,54,17,58},

{19,80,81,68,5,94,47,69,28,73,92,13,86,52,17,77,4,89,55,40},

{4,52,8,83,97,35,99,16,7,97,57,32,16,26,26,79,33,27,98,66},

{88,36,68,87,57,62,20,72,3,46,33,67,46,55,12,32,63,93,53,69},

{4,42,16,73,38,25,39,11,24,94,72,18,8,46,29,32,40,62,76,36},

{20,69,36,41,72,30,23,88,34,62,99,69,82,67,59,85,74,4,36,16},

{20,73,35,29,78,31,90,1,74,31,49,71,48,86,81,16,23,57,5,54},

{1,70,54,71,83,51,54,69,16,92,33,48,61,43,52,1,89,19,67,48},

};

unsigned m = 0;

unsigned i, j;

for (i = 0; i < N-3; i++) {

for (j = 0; j < N-3; j++) {

unsigned h = grid[i][j] \* grid[i][j+1] \* grid[i][j+2] \* grid[i][j+3];

unsigned v = grid[j][i] \* grid[j+1][i] \* grid[j+2][i] \* grid[j+3][i];

unsigned d1 = grid[i][j] \* grid[i+1][j+1] \* grid[i+2][j+2] \* grid[i+3][j+3];

unsigned d2 = grid[i][N-j-1] \* grid[i+1][N-j-2] \* grid[i+2][N-j-3] \* grid[i+3][N-j-4];

m = max(m, max(h, max(v, max(d1, d2))));

}

}

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

return 0;

}

unsigned max(unsigned a, unsigned b)

{

return a > b ? a : b;

}

14. The following iterative sequence is defined for the set of positive integers: n → n/2 (n is even) n → 3n + 1 (n is odd)

Using the rule above and starting with 13, we generate the following sequence: 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1 It can be seen that this sequence (starting at 13 and finishing at 1) contains 10 terms. Although it has not been proved yet (Collatz Problem), it is thought that all starting numbers finish at 1. Which starting number, under one million, produces the longest chain? Ans: 837799

#include <stdio.h>

static unsigned collatz\_count(unsigned n);

int main(void)

{

unsigned i, max\_c = 0, max\_i = 0;

for (i = 1; i < 1000000; i++) {

unsigned c = collatz\_count(i);

if (c > max\_c) {

max\_c = c;

max\_i = i;

}

}

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

return 0;

}

unsigned collatz\_count(unsigned n)

{

unsigned c = 0;

while (n > 1) {

n = n%2==0 ? n/2 : 3\*n+1;

c++;

}

return c+1;

}

18. By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

3 7 4 2 4 6 8 5 9 3 That is, 3 + 7 + 4 + 9 = 23.

Find the maximum total from top to bottom of the triangle below: 75 95 64 17 47 82 18 35 87 10 20 04 82 47 65 19 01 23 75 03 34 88 02 77 73 07 63 67 99 65 04 28 06 16 70 92 41 41 26 56 83 40 80 70 33 41 48 72 33 47 32 37 16 94 29 53 71 44 65 25 43 91 52 97 51 14 70 11 33 28 77 73 17 78 39 68 17 57 91 71 52 38 17 14 91 43 58 50 27 29 48 63 66 04 68 89 53 67 30 73 16 69 87 40 31 04 62 98 27 23 09 70 98 73 93 38 53 60 04 23 Ans: 1074

#include <stdio.h>

#define N 15

static int findmax(const char triangle[][N], int row, int col);

int main(void)

{

const char triangle[N][N] = {

{75},

{95,64},

{17,47,82},

{18,35,87,10},

{20, 4,82,47,65},

{19, 1,23,75, 3,34},

{88, 2,77,73, 7,63,67},

{99,65, 4,28, 6,16,70,92},

{41,41,26,56,83,40,80,70,33},

{41,48,72,33,47,32,37,16,94,29},

{53,71,44,65,25,43,91,52,97,51,14},

{70,11,33,28,77,73,17,78,39,68,17,57},

{91,71,52,38,17,14,91,43,58,50,27,29,48},

{63,66, 4,68,89,53,67,30,73,16,69,87,40,31},

{ 4,62,98,27,23, 9,70,98,73,93,38,53,60, 4,23},

};

printf("%d\n", findmax(triangle, 0, 0));

return 0;

}

int findmax(const char triangle[][N], int row, int col)

{

int left, right;

if (row == N) {

return triangle[row][col];

}

left = findmax(triangle, row+1, col);

right = findmax(triangle, row+1, col+1);

return triangle[row][col] + (left > right ? left : right);

}