Project Euler

Question 1.

#include<stdio.h>

int main()

{

int i;

int sum =0;

for(i=2;i>=1000;i++)

{

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

{

sum = sum+i;

}

}

return 0;

}

Question 2.

#include<stdio.h>

int main()

{

int i;

int first =0;

int second =1;

int next;

for(i=0;i>4000000;i++)

{

next = first+second;

first = second;

second = next;

if(next%2==0)

{

sum = sum+i;

}

}

return 0;

}

Question no 3

#include<stdio.h>

{

int i,j;

int sum =0;

int largest =2;

while(n%2==0)

{

n=n/2;

}

for(i=3;i=>sqrt(n);i=i+2)

{

while(n%i==0)

{

n=n%i;

}

If(n>largest)

{

largest=n;

}

}

printf(“%d”,largest=n);

}

Question no.4

#include<stdio.h>

int main()

{

int i

int product;

int largest;

largest =2;

for(i=100;1>=999;i=i+1)

{

Product =i\*(i+1);

result = palindrome(product);

if(result ==1)

{

If(product>largest)

{

largest=product;

}

}

}

return 0;

int palindrome(int product)

{

int t;

t=product;

int r=0;

while(t>0)

{

r=r\*10+t%10;

t=t/10;

}

If(product==r)

{

return 1;

}

else

{

return 0;

}

}

Question no. 5

#include<stdio.h>

int main()

{

int x,y,smallest;

x=1;

y=1;

smallest =0;

while (x>0)

{

y=1;

while(x%y==0)&&(y<=0)

{

If(y==20)

{

Smallest=x;

break ;

}

y ++;

}

x ++;

}

}

Question no 6.

#include<stdio.h>

int main()

{

int i,j;

int sum1=0;

int sum2=0;

int result;

for(i=0;i<100;i++)

{

sum 1 = sum1+(i\*i);

}

for(j=0;j<100;j++)

{

sum 2=sum2+j;

}

sum 2 =sum2\*sum2;

result = sum2-sum1;

printf(“%d”,result);

return 0;

}

Question no. 7

#include<stdio.h>

int main()

{

int i,j;

int count =1;

int flag =1;

int n=2;

while(count !=10001)

{

for(i=1;i<=sqrt(n);i++)

{

if((n%i)==0))

{

flag=1;

break;

}

}

if(flag==1)

{

Count =count+1;

}

n=n+1;

}

return 0;

}

Question no .8

#include<stdio.h>

int main()

{

char\*num[]="73167176531330624919225119674426574742355349194934

96983520312774506326239578318016984801869478851843

85861560789112949495459501737958331952853208805511

12540698747158523863050715693290963295227443043557

66896648950445244523161731856403098711121722383113

62229893423380308135336276614282806444486645238749

30358907296290491560440772390713810515859307960866

70172427121883998797908792274921901699720888093776

65727333001053367881220235421809751254540594752243

52584907711670556013604839586446706324415722155397

53697817977846174064955149290862569321978468622482

83972241375657056057490261407972968652414535100474

82166370484403199890008895243450658541227588666881

16427171479924442928230863465674813919123162824586

17866458359124566529476545682848912883142607690042

24219022671055626321111109370544217506941658960408

07198403850962455444362981230987879927244284909188

84580156166097919133875499200524063689912560717606

05886116467109405077541002256983155200055935729725

71636269561882670428252483600823257530420752963450";

int i ;product=0;final\_product=0;

for( i=0;num[i]!=NULL;i++)

{

printf(“%i\n”,num[i]-‘0’);

temp = temp\*num[i]-‘0’;

if((i+1)%13==0)

{

product = temp;

temp=1;

if(product>final\_product)

{

final \_product = product;

}

return 0;

}

Question no. 9

#include<stdio.h>

#include<math.h>

int main()

{

int a,b;

double c ,c\_squared,p;

for(a=1;a<1000;a++)

{

for(b=a+1;b<=1000;b++)

{

c\_squared =pow(a,2)+pow(b,2);

c=pow(c\_squared,0.5);

if(a+b+c==1000)

{

p=a\*b\*c;

break;

}

}

}

return 0;

}

Question no. 10

#include<stdio.h>

#include<math.h>

int main()

{

long long number,sum;

int flag =0;

sum=0;

number =0;

int i;

double boundary;

while(number<2000000)

{

flag=0;

if(number==1)

{

flag=1;

}

if(number%2==0)

{

flag=1;

}

if(number==2||number=3)

{

flag=0;

}

boundary=(int)floor(sqrt(number));

for(i=3;i<=boundary;i+=2)

{

if(number%i==0)

{

flag=1;

}

if(flag==0)

{

sum+=number;

}

number++;

}

}