//By Ian Whittemore

#include "stdafx.h"

#include <iostream>

using namespace std;

int divisors(int n);

void dpa(int n);

int xnorOp(int a, int b);

int reversednumber(int p);

int main()

{

int option;

do

{

cout << "Please enter an option \n 1: The sum of all divisors and the type it is \n 2: The xnor of two binary numbers \n 3: Reverse the digits \n 4: To end program" << endl;

cin >> option;

switch (option)

{

case 1:

{

int n;

cout << "Please enter a number to work out of the sum of divisors" << endl;

cin >> n;

cout << "The sum of the divisors are : " << divisors(n) << endl;

dpa(n);

}

break;

case 2:

{

int a, b;

cout << "Please enter a binary number " << endl;

cin >> a;

cout << "Please enter another binary number " << endl;

cin >> b;

cout << a << " XNOR " << b << " = " << xnorOp(a, b) << endl;

}

break;

case 3:

{

int p;

cout << "Please enter an integer " << endl;

cin >> p;

cout << "The reveresed number is : " << reversednumber(p) << endl;

}

break;

case 4:

{

cout << "Thank you for using my program \n By Ian Whittemore \n ID : 1700870" << endl;

exit(0);

}

break;

default:

break;

}

} while (option <= 4);

cout << "Thank you for using my program \n By Ian Whittemore \n ID : 1700870" << endl;

return 0;

}

/\*

Pseudo Code

Begin

variable divisor, result

do

input divisor

if(n%i == 0)

if (I = divisor / i)

divisor += i

else

result += I + divisor / I

end loop

result = result + 1

output result

End

\*/

int divisors(int n)

{

int result = 0;

for(int i = 2; i <= sqrt(n); i++)

{

if (n % i == 0)

{

if (i == (n / i))

{

result += i;

}

else

{

result += (i + n / i);

}

}

}

result = result + 1;

return result;

}

/\*

Pseudo Code

Begin

variable divisor, result

do

input divisor

while (divisor < 0)

for loop

if (divisor % i ==0)

if (I = divisor /i)

divisor += i

else

result += I + divisor/I

end loop

result = result + 1

if (result > divisor)

Output: Abundant

Else if (result < divisor)

Output: Deficient

Else

Output Perfect

End

\*/

void dpa(int n)

{

if (divisors(n) < n)

{

cout << "Deficient " << endl;

}

else if (divisors(n) == n)

{

cout << "Perfect " << endl;

}

else if (divisors(n) > n)

{

cout << "Abundant " << endl;

}

}

/\*

Pseudo Code

Begin

variable sum,digit = 1,result

while (a > 0 && b > 0)

{

if (a = 1 , b =1 , b = 0, b = 0)

{

result = 1;

}

else

{

result = 0;

}

sum = result \* digit + sum;

digit = digit \* 10;

a / 10 , b / 10

}

return sum;

Output xnor

\*/

int xnorOp(int a, int b)

{

int sum = 0;

int digit = 1;

int result;

while (a > 0 && b > 0)

{

if (a % 10 == 1 && b % 10 == 1 || a % 10 == 0 && b % 10 == 0)

{

result = 1;

}

else

{

result = 0;

}

sum = result \* digit + sum;

digit = digit \* 10;

a = a / 10;

b = b / 10;

}

return sum;

}

/\*

Pseudo Code

Begin

variable num,remainder

while p doesnt = 0

reaminder = p % 10

num = num \* 10 + remainder

p = p / 10

End loop

Output: reversed number

End

\*/

int reversednumber(int p)

{

int number = 0;

int remainder;

while(p != 0)

{

remainder = p % 10;

number = number \* 10 + remainder;

p /= 10;

}

return number;

}