**算法训练 最大最小公倍数**

时间限制：1.0s   内存限制：256.0MB

问题描述

已知一个正整数N，问从1~N中任选出三个数，他们的最小公倍数最大可以为多少。

输入格式

输入一个正整数N。

输出格式

输出一个整数，表示你找到的最小公倍数。

样例输入

9

样例输出

504

数据规模与约定

1 <= N <= 106。

锦囊1

使用贪心来选择。

锦囊2

当n为奇数时，答案一定是n\*(n-1)\*(n-2)。

当n为偶数时，答案可能是(n-1)\*(n-2)\*(n-3)，也可能是n\*a\*b，其中a>=n-3。

本题的C++参考代码如下：

#include<iostream>

using namespace std;

//两个数的最大公约数

long long gcd(long long a,long long b)

{

return b?gcd(b,a%b):a;

}

int main()

{

long long n,a,b,r;

while(cin>>n)

{

if(n%2==1)

cout<<n\*(n-1)\*(n-2)<<endl;

else

{

long long s1=(n-1)\*(n-2)\*(n-3);

a=n\*(n-1);

while(gcd(n-2,a)!=1)n--;

if((n-2)\*a>s1)

cout<<(n-2)\*a<<endl;

else

cout<<s1<<endl;

}

}

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 10000

//输出值

void Print(char \*num,int n)

{

int i;

for(i=n-1;i>=0;i--)

{

printf("%c",num[i]);

}

printf("\0");

printf("\n");

}

//对字符数组进行初始化

void Init(char \*num)

{

int i;

for(i=0;i<MAXSIZE;i++) num[i]='0';

}

//将一个整数转换为字符类型

int TransformString(char \*num,int value)

{

int num\_n=0;

while(value)

{

num[num\_n++]=value%10+'0';

value/=10;

}

return num\_n;

}

//两个大整数相乘

int Multiplication(char \*num1,int num1\_n,char \*num2,int num2\_n,char \*result,int result\_n)

{

int i,j;

int carry;

int value;

for(i=0;i<num2\_n;i++)

{

carry=0;

for(j=0;j<num1\_n;j++)

{

value=(num2[i]-'0')\*(num1[j]-'0')+(result[i+j]-'0')+carry;

carry=value/10;

result[i+j]=value%10+'0';

}

if(carry)

{

result[i+j]=carry+'0';

result\_n=i+j+1;

}

else result\_n=i+j;

}

return result\_n;

}

//判断两个数是否有公约数

int GCD(int num1,int num2)

{

int temp;

while(num2)

{

temp=num1%num2;

num1=num2;

num2=temp;

}

if(num1==1) return 1;

else return 0;

}

//选出满足条件的值

void Select(int n)

{

char num1[MAXSIZE];

char num2[MAXSIZE];

char result[MAXSIZE];

int num1\_n;

int num2\_n;

int result\_n=0;

int n1;

int n2;

int n3;

if(n%2==1)

{

n1=n;

n2=n-1;

n3=n-2;

}

else

{

n1=n;

n2=n-1;

n3=n-3;

while( GCD(n1,n3)!=1 || GCD(n2,n3)!=1) n3--;

if(n3!=n-3)

{

n1=n-1;

n2=n-2;

n3=n-3;

}

}

num1\_n=TransformString(num1,n1);

num2\_n=TransformString(num2,n2);

Init(result);

result\_n=Multiplication(num1,num1\_n,num2,num2\_n,result,result\_n);

num1\_n=TransformString(num1,n3);

Init(num2);

num2\_n=0;

num2\_n=Multiplication(num1,num1\_n,result,result\_n,num2,num2\_n);

Print(num2,num2\_n);

}

int main()

{

int n;

scanf("%d",&n);

Select(n);

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

public class Main{

public static long gcdivisor(long a,long b)

{

long m=1;

if(a<b)

{

m=a;

a=b;

b=m;

}

while(m!=0)

{

m=a%b;

a=b;

b=m;

}

return a;

}

public static void main(String args[]) throws IOException{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

long n=Long.parseLong(br.readLine());

if(n%2==1)

System.out.println(n\*(n-1)\*(n-2));

else

{

if(gcdivisor(n,n-3)!=1)

System.out.println((n-1)\*(n-2)\*(n-3));

else

{

System.out.println(n\*(n-1)\*(n-3));

}

}

}

}