问题描述

　　输入一个正整数*n*，输出*n*!的值。  
　　其中*n*!=1\*2\*3\*…\**n*。

算法描述

*n*!可能很大，而计算机能表示的整数范围有限，需要使用高精度计算的方法。使用一个数组*A*来表示一个大整数*a*，*A*[0]表示*a*的个位，*A*[1]表示*a*的十位，依次类推。  
　　将*a*乘以一个整数*k*变为将数组*A*的每一个元素都乘以*k*，请注意处理相应的进位。  
　　首先将*a*设为1，然后乘2，乘3，当乘到*n*时，即得到了*n*!的值。

输入格式

　　输入包含一个正整数*n*，*n*<=1000。

输出格式

　　输出*n*!的准确值。

样例输入

10

样例输出

3628800

锦囊1

数组。

锦囊2

使用数组来保存一个整数，按手算的方法处理。

import java.math.BigInteger;

import java.util.\*;

public class Main{

protected static ArrayList table = new ArrayList();

static {

table.add(BigInteger.valueOf(1));

}

public static synchronized BigInteger factorial(int x) {

for (int size = table.size(); size <= x; size++) {

BigInteger lastfact = (BigInteger) table.get(size - 1);

BigInteger nextfact = lastfact.multiply(BigInteger.valueOf(size));

table.add(nextfact);

}

return (BigInteger) table.get(x);

}

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int n=s.nextInt();

System.out.print(factorial(n));

}

}

#include <stdio.h>

#define N 10000

int main()

{

int a[N]={1};

int k=0,l=1,n;

int i,j;

scanf("%d",&n);

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

{

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

{

a[j]=a[j]\*i+k;

k=a[j]/10000;

a[j]=a[j]%10000;

}

if(k)

{

a[j]=k;

l++;

k=0;

}

}

printf("%d",a[l-1]);

for(i=l-2;i>=0;i--)

printf("%04d",a[i]);

printf("\n");

return 0;

}

#include <stdio.h>

#include <string.h>

#define MAX 10000

#define mod 10000

#define baselen 4

#define in(a) scanf("%d",&a)

#define out1(a) printf("%d",a)

#define out2(a) printf("%04d",a)

typedef int type;

struct bint{

type dig[MAX], len;

bint(){len = 0, dig[0] = 0;}

};

void by(bint a, type b, bint& c){

type i, carry;

for( i = carry = 0; i <= a.len || carry; i++){

if( i <= a.len ) carry += b\*a.dig[i];

c.dig[i] = carry%mod;

carry /= mod;

}

i--;

while( i && !c.dig[i] )i--;

c.len = i;

}

bool input(bint& a){

type i, j, w, k, p;

char data[MAX\*baselen+1];

if(scanf("%s",data)==EOF)return false;

w = strlen(data) - 1, a.len = 0;

for(p=0;p<=w&&data[p]=='0';p++);

while(1){

i = j = 0, k = 1;

while(i<baselen&&w>=p){

j = j+ (data[w--] - '0')\*k;

k \*= 10, i++;

}

a.dig[a.len++] = j;

if(w<p)break;

}

a.len--;

return true;

}

void output(bint& a){

type i;

i = a.len - 1;

out1(a.dig[a.len]);

while(i>=0)out2(a.dig[i--]);

}

void give(type a, bint& b){

b.dig[0] = a%mod;

a /= mod;

if(a>0)b.dig[1] = a, b.len = 1;

else b.len = 0;

}

int main()

{

bint a;int b,i;scanf("%d",&b);give(1,a);

for(i=2;i<=b;i++)by(a,i,a);

output(a);printf("\n");

return 0;

}