|  |
| --- |
| You are given the ints **n**, **K**, and **v**. Consider the following modular equation with **n** variables: (x[0] \* x[1] \* x[2] \* ... \* x[**n**-1]) mod **K** = **v**. How many solutions does it have?  Formally, we want to find the number of sequences (x[0], x[1], ..., x[**n**-1]) such that each x[i] is an integer between 0 and **K**-1, inclusive, and the product of all x[i] gives the remainder **v** when divided by **K**. Please compute and return the number of such sequences, modulo 10^9+7.  //CODE  struct info  {  LL mat[105][105];  }aa,bb,cc,res;  info foo(info aa,info bb,LL m)  {  rep(i,m)rep(j,m)  {  LL sum=0;  rep(k,m)  sum+=(aa.mat[i][k]\*bb.mat[k][j]), sum%=MOD;  res.mat[i][j]=sum;  }  return res;  }  LL fo(info aa,info bb,LL n)  {  while(n)  {  if(n%2==1)aa=foo(aa,bb,m);  bb=foo(bb,bb,m);  n/=2;  }  rep(i,m)  rr+=aa.mat[i][r],rr%=MOD;  return rr;  }  int main()  {  cin>>n>>m>>r;  rep(i,m) rep(j,m)bb.mat[i][(i\*j)%m]++;  rep(i,m) rep(j,m) if(i==j)aa.mat[i][j]=1;  LL ans=fo(aa,bb,n-1); cout<<ans<<endl;  } |