Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:01:29 **Register Number:** RA2031241010094

Q. DT-11:Celsius To Farenheit

Mercy and her friends went a trip to hill station during winter. It is raining heavily so they cant go outside. mercy felt very bored and decided to ask puzzle to her friends.
She gave them the temperature in Celsius format and asked them to convert it to Fahrenheit

Source Code

```
#include <stdio.h>
int main()
{
  float c,f;
  scanf("%f",&c);
  f=(c*9/5)+32;
  printf("%.2f",f);
  printf(" Fahrenheit");
  return 0;
}
```

Sample Input

45.0

Sample Output

113.00 Fahrenheit

Result

Thus, Program " **DT-11:Celsius To Farenheit** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:02:10 **Register Number:** RA2031241010094

Q. Star formation

Pyramid Star Formation

Source Code

```
#include <stdio.h>
int main()
{
  int i,j,n;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=i;j++)
    {
      printf("*");
    }
    printf("\n");
  }
  return 0;
}
```

Sample Input

5

Sample Output

```
*
**
***
****
*****
```

Result

Thus, Program " **Star formation** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:02:39 **Register Number:** RA2031241010094

Q. GCD and LCM

Two integers A and B are the inputs. Write a program to find GCD and LCM of A and B.
Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer A and B.
Output

Display the GCD and LCM of A and B separated by space respectively.
Constraints

1 <= T <= 1000
1 <= A,B <= 1000000

Source Code

```
#include <stdio.h>
int main()
{
  int num,den,rem,gcd,lcm,a,b,n,i;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  {
      scanf("%d",&num);
      scanf("%d",&den);
      if(num>den)
       {
         a=num;
         b=den;
       }
      else
       {
        a=den;
        b=num;
       }
      rem=a%b;
      while(rem!=0)
       {
        a=b;
        b=rem;
        rem=a%b;
       }
      gcd=b;
      lcm=num*den/gcd;
printf("%d",gcd);
printf(" %d\n",lcm);
}
  return 0;
}
```

Sample Input

```
5
2 3
2 4
3 5
4 6
7 8
```

Sample Output

```
1 6
2 4
1 15
2 12
1 56
```

Result

Thus, Program " **GCD and LCM** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:03:07 **Register Number:** RA2031241010094

Q. Financial Crisis

Rishab owns a rectangular plot of land . Due to certain financial problems , he wishes to sell some parts of his plot. Rishab finds out that square shaped plots sell better than rectangular ones . With this in mind, he decides to divide his plot into minimum possible square plots so that he can get maximum profit . All the square plots have the same dimension.

Given the dimensions of his plot , Write a Program to calculate the minimum number of square plots.

Input
The first line of input consists of two integers L and B which denotes the length and breadth of the rectangular plot.

Output
Output is a single line which denotes the minimum number of square plots that can be formed

Source Code

```
#include <stdio.h>
int main()
{
  int i,j,l,b,x,a,c;
  for(i=1;i<=1;i++)
  {
    scanf("%d%d",&l,&b);
    for(j=1;j<=l&&j<=b;j++)
    {
      if(l%j==0&&b%j==0)
      {
        x=j;
      }
    }
    a=l/x;
    c=b/x;
    printf("%d",(a*c));
  }
 return 0;
}
```

Sample Input

4 6

Sample Output

6

Result

Thus, Program " **Financial Crisis** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:03:35 **Register Number:** RA2031241010094

Q. Prime Factors

Helan had to find a number that must be the sum of two prime numbers. Help him to write a C Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers

Source Code

```
#include <stdio.h>
int isprime(int n);
int main()
{
  int i,n,flag=0;
  scanf("%d",&n);
  for(i=2;i<=n/2;i++)
  {
     if(isprime(i)==1)
     {
       if(isprime(n-i)==1)
       {
         printf("%d = %d + %d\n",n,i,n-i);
         flag=1;
       }
     }
  }
  if(flag==0)
    printf("NOT");
  return 0;
}
int isprime(int n)
{
  int i,ip=1;
  for(i=2;i<=n/2;i++)
  {
    if(n%i==0)
    {
      ip=0;
      break;
    }
  }
  return ip;
}
```

Sample Input

35

Sample Output

NOT

Result

Thus, Program " **Prime Factors** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:04:09 **Register Number:** RA2031241010094

Q. read data on a regular basis

"Alexey is trying to develop a program for a very simple microcontroller. It makes readings from various sensors over time, and these readings must happen at specific regular times. Unfortunately, if two of these readings occur at the same time, the microcontroller freezes and must be reset.
There are N different sensors that read data on a regular basis. For each i from 1 to N, the reading from sensor i will occur every Ai milliseconds with the first reading occurring exactly Ai milliseconds after the microcontroller is powered up. Each reading takes precisely one millisecond on Alexey's microcontroller.
Alexey wants to know when the microcontroller will freeze after he turns it on.
Input
The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.
The first line contains single integer N denoting the number of sensors.
The second line contains N space-separated integers A1, A2, ..., AN denoting frequency of measurements. Namely, sensor i will be read every Ai milliseconds with the first reading occurring Ai milliseconds after the microcontroller is first turned on.
Output
For each test case, output a single line containing the number of milliseconds until the microcontroller freezes.

Constraints
1< T< 10
2< N < 500

1< Ai < 109"

Source Code

```
#include <stdio.h>
int gcd(int u,int v)
  {
    int t,a=u,b=v,ans;
  while(u>0)
  {
    if(u<v)
    {
      t=u;
      u=v;
      v=t;
    }
    u=u-v;
  }
  ans=(a*b)/v;
  return ans;
}
int main()
{
  int t,z;
  scanf("%d",&t);
  for(z=0;z<t;z++)
  {
    int n,i,j;
    scanf("%d",&n);
    int num[n],ans=10000000;
    for(i=0;i<n;i++)
    {
      scanf("%d",&num[i]);
    }
    for(i=0;i<n;i++)
    {
      for(j=i+1;j<n;j++)
      {
        int temp=gcd(num[i],num[j]);
        if(temp<ans)
          ans=temp;
      }
    }
    printf("%d\n",ans);
  }
  return 0;
}
```

Sample Input

```
3
3
2 3 5
4
1 8 7 11
4
4 4 5 6
```

Sample Output

```
6
7
4
```

Result

Thus, Program " **read data on a regular basis** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:04:38 **Register Number:** RA2031241010094

Q. Modulo of numbers(Integer data type)

Jennys home work for Fifth day is to find modulo of two numbers, help jenny to solve the problem.

Source Code

```
#include <stdio.h>
int main()
{
  int n,n1,rem;
  scanf("%d%d",&n,&n1);
  rem=n%n1;
  printf("The reminder of two number is:%d",rem);
  return 0;
}
```

Sample Input

```
6
2
```

Sample Output

```
The reminder of two number is:0
```

Result

Thus, Program " **Modulo of numbers(Integer data type)** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:05:15 **Register Number:** RA2031241010094

Q. Check the Scientist

Bogar (one of the 18 Tamil Siddhars) was one of the great scientist. The Indian council decided that we need to assign some number as a gift to the great scientist.
There was a suggestion given by the Indian Council. If the sum of cube of each number is again equal to the number then they decided that they can assign the number to the great scientist.
Kindly help the Indian Council to complete the task by writing a simple logic.

Source Code

```
#include <stdio.h>
int main()
{
  int n,n1,dig,sum=0;
  scanf("%d",&n);
  n1=n;
  while(n1!=0)
  {
    dig=n1%10;
    sum=sum+dig*dig*dig;
    n1=n1/10;
  }
  if(sum==n)
    printf("Give to Scientist Bogar");
  else
    printf("Dont Give to Scientist Bogar");
return 0;
}
```

Sample Input

531

Sample Output

Dont Give to Scientist Bogar

Result

Thus, Program " **Check the Scientist** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:05:44 **Register Number:** RA2031241010094

Q. Narcissistic raju

Help for Raju to Check the given number is Narcissistic Number. It is similar to an Armstrong Number.If the Sum of Digits of a Number raised to the power of the number of digits is equal to the Number/Integer, then it is a Narcissistic Number.

Source Code

```
#include <stdio.h>
#include<math.h>
int main()
{
  int n,n1,p=0,dig,sum=0;
  scanf("%d",&n);
  n1=n;
  while(n1!=0)
  {
    n1=n1/10;
    p++;
  }
 n1=n;
  while(n>0)
  {
    dig=n%10;
    sum=sum+pow(dig,p);
    n=n/10;
  }
  if(sum==n1)
    printf("Narcissistic Number");
  else
 printf("NOT Narcissistic Number");
 return 0;
}
```

Sample Input

153

Sample Output

Narcissistic Number

Result

Thus, Program " **Narcissistic raju** " has been successfully executed

Course: C **Session:** Data types **Timestamp:** 2021-1-9 18:06:17 **Register Number:** RA2031241010094

Q. Magical game of sum

Yesterday, puppy Tuzik learned a magically efficient method to find the sum of the integers from 1 to N. He denotes it as sum(N). But today, as a true explorer, he defined his own new function: sum(D, N), which means the operation sum applied D times: the first time to N, and each subsequent time to the result of the previous operation.

For example, if D = 2 and N = 3, then sum(2, 3) equals to sum(sum(3)) = sum(1 + 2 + 3) = sum(6) = 21.

Tuzik wants to calculate some values of the sum(D, N) function. Will you help him with that?

Input

The first line contains a single integer T, the number of test cases. Each test case is described by a single line containing two integers D and N.

Output

For each testcase, output one integer on a separate line.

Constraints

1 < T< 16

1

Source Code

```
#include <stdio.h>
int sum1(int d);
int sum(int a,int b);
int main()
{
  int q,e,c,pupy;
  scanf("%d\n",&q);
  while(q>0)
  {
    scanf("%d %d\n",&e,&c);
    pupy=sum(e,c);
    printf("%d\n",pupy);
          q--;
  }
}
int sum1(int d)
  {
    int s=0,i;
    for(i=0;i<=d;i++)
    {
      s=s+i;
    }
    return s;
  }
int sum(int a,int b)
  {
    int p=0;
    while(a>0)
    {
      p=sum1(b);
      a--;
      b=p;
    }
    return p;
  }
```

Sample Input

```
2
1 4
2 3
```

Sample Output

```
10
21
```

Result

Thus, Program " **Magical game of sum** " has been successfully executed