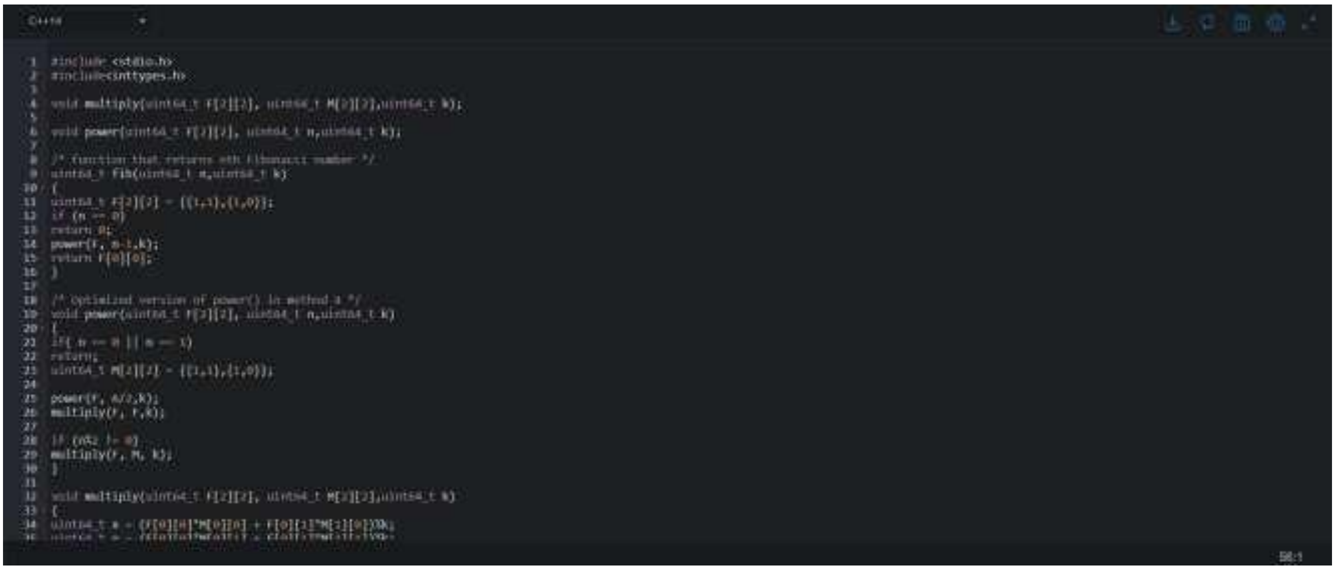
WORKSHEET 1

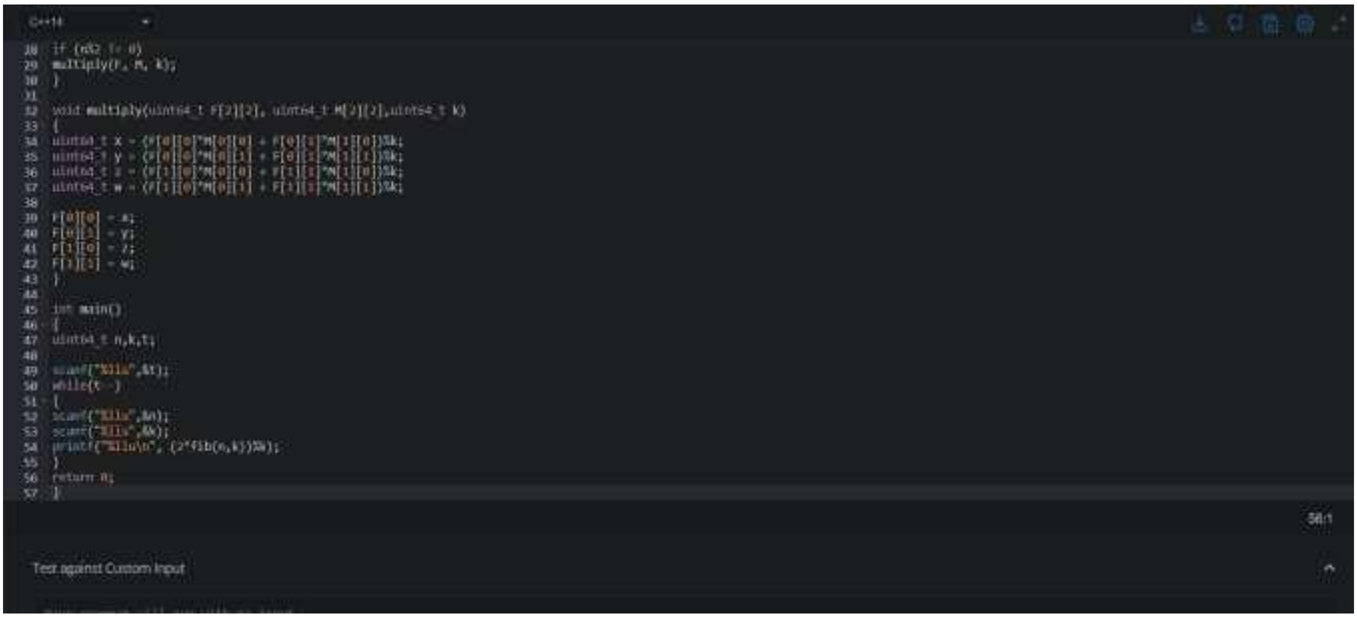
NAME- EKROOP SINGH	SEC-DWWC 43
UID:21BCS8143	Date- 03/01/2023

1. **Fire and Ice**

Program Code:

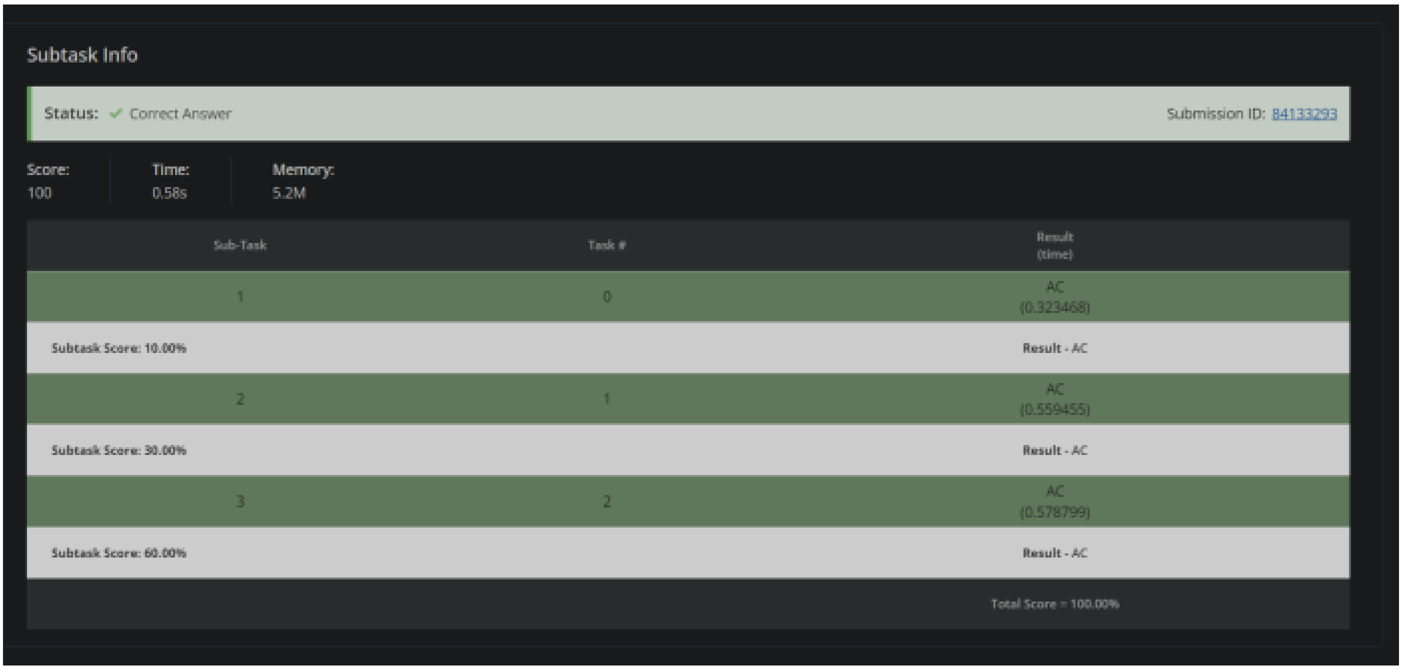


```
#include <stdio.h>
#include<inttypes.h>

void multiply(uint64_t F[2][2], uint64_t M[2][2],uint64_t k);

void power(uint64_t F[2][2], uint64_t n,uint64_t k);

/* function that returns nth Fibonacci number */
uint64_t fib(uint64_t n,uint64_t k)
{
uint64_t F[2][2] = {{1,1},{1,0}};
if (n == 0)
return 0;
power(F, n-1,k);
return F[0][0];
}

/* Optimized version of power() in method 4 */
void power(uint64_t F[2][2], uint64_t n,uint64_t k)
{
if( n == 0 || n == 1)
return;
uint64_t M[2][2] = {{1,1},{1,0}};

power(F, n/2,k);
multiply(F, F,k);

if (n%2 != 0)
multiply(F, M, k);
}

void multiply(uint64_t F[2][2], uint64_t M[2][2],uint64_t k)
{
uint64_t x = (F[0][0]*M[0][0] + F[0][1]*M[1][0])%k;
```



```
if (n%2 != 0)
multiply(F, M, k);
}

void multiply(uint64_t F[2][2], uint64_t M[2][2],uint64_t k)
{
uint64_t x = (F[0][0]*M[0][0] + F[0][1]*M[1][0])%k;
uint64_t y = (F[0][0]*M[0][1] + F[0][1]*M[1][1])%k;
uint64_t z = (F[1][0]*M[0][0] + F[1][1]*M[1][0])%k;
uint64_t w = (F[1][0]*M[0][1] + F[1][1]*M[1][1])%k;

F[0][0] = x;
F[0][1] = y;
F[1][0] = z;
F[1][1] = w;
}

int main()
{
uint64_t n,k,t;

scanf("%llu",&t);
while(t--)
{
scanf("%llu",&n);
scanf("%llu",&k);
printf("%llu\n", (2*fib(n,k))%k);
}
return 0;
}
```

Output:



Subtask Info

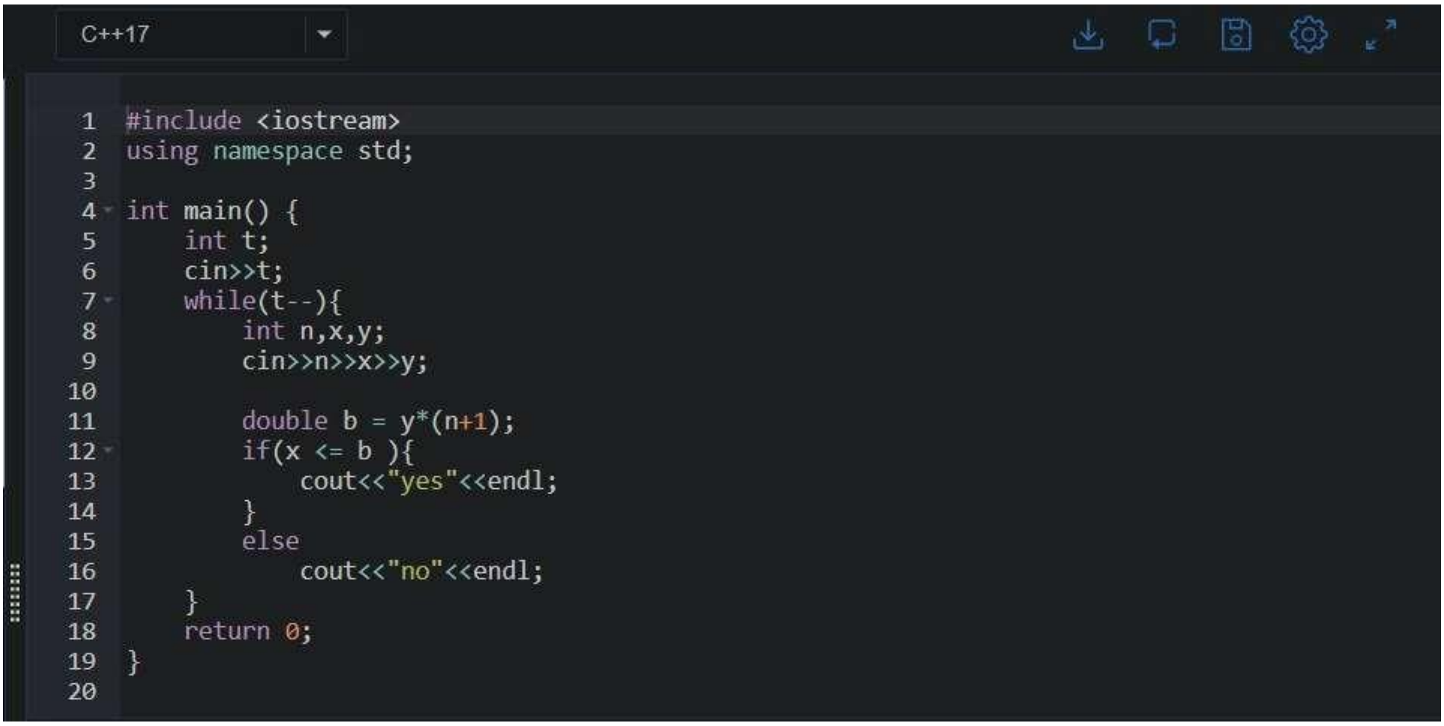
Status: ✓ Correct Answer   Submission ID: 84133293

Score: 100 | Time: 0.58s | Memory: 5.2M

Sub-Task	Task #	Result (time)
1	0	AC (0.323468)
Subtask Score: 10.00%		Result - AC
2	1	AC (0.559455)
Subtask Score: 30.00%		Result - AC
3	2	AC (0.578799)
Subtask Score: 60.00%		Result - AC
		Total Score = 100.00%

2. Gold Mining

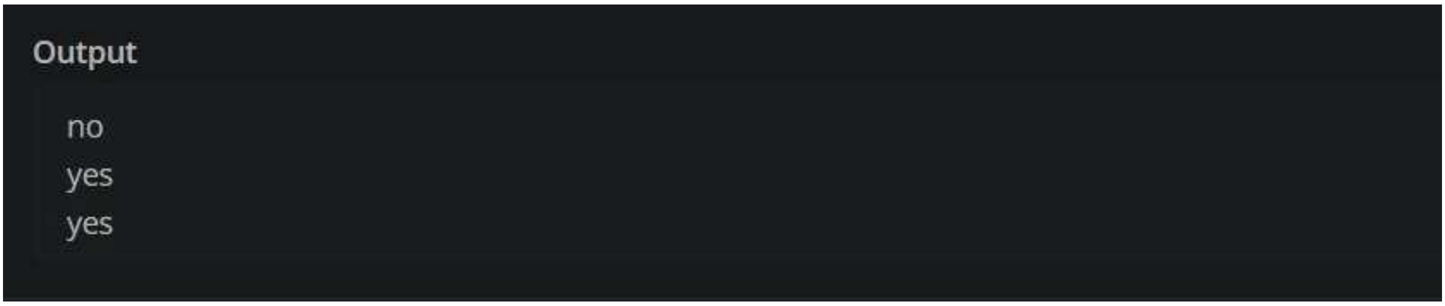
Program Code:



C++17

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--){
        int n,x,y;
        cin>>n>>x>>y;

        double b = y*(n+1);
        if(x <= b ){
            cout<<"yes"<<endl;
        }
        else
            cout<<"no"<<endl;
    }
    return 0;
}
```

<u>Output</u>

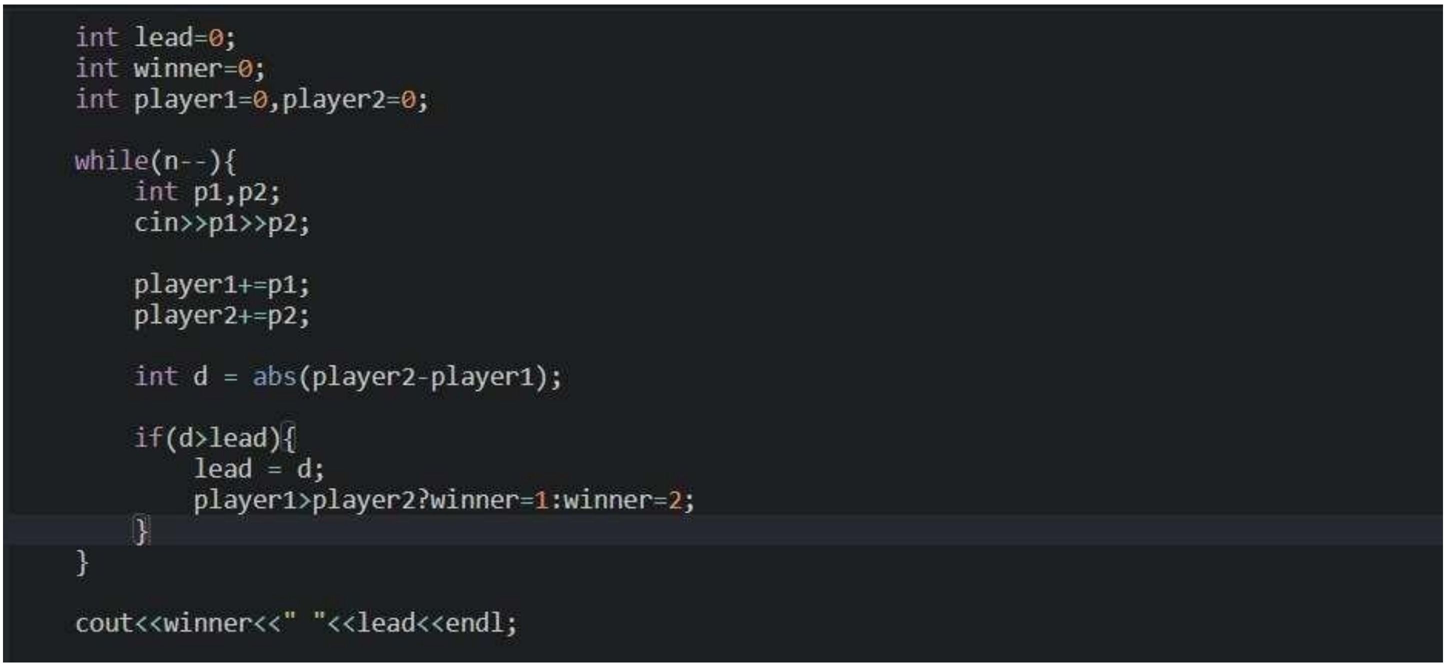


Output

```
no
yes
yes
```

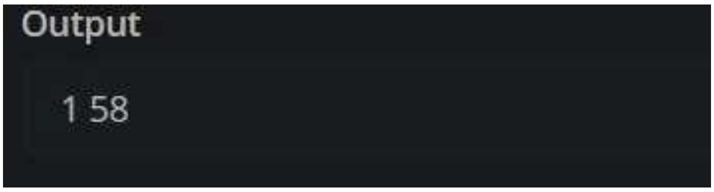
3. The Lead Game

Program Code:



```
int lead=0;
int winner=0;
int player1=0,player2=0;

while(n--){
    int p1,p2;
    cin>>p1>>p2;

    player1+=p1;
    player2+=p2;

    int d = abs(player2-player1);

    if(d>lead){
        lead = d;
        player1>player2?winner=1:winner=2;
    }
}

cout<<winner<<" "<<lead<<endl;
```

Output:

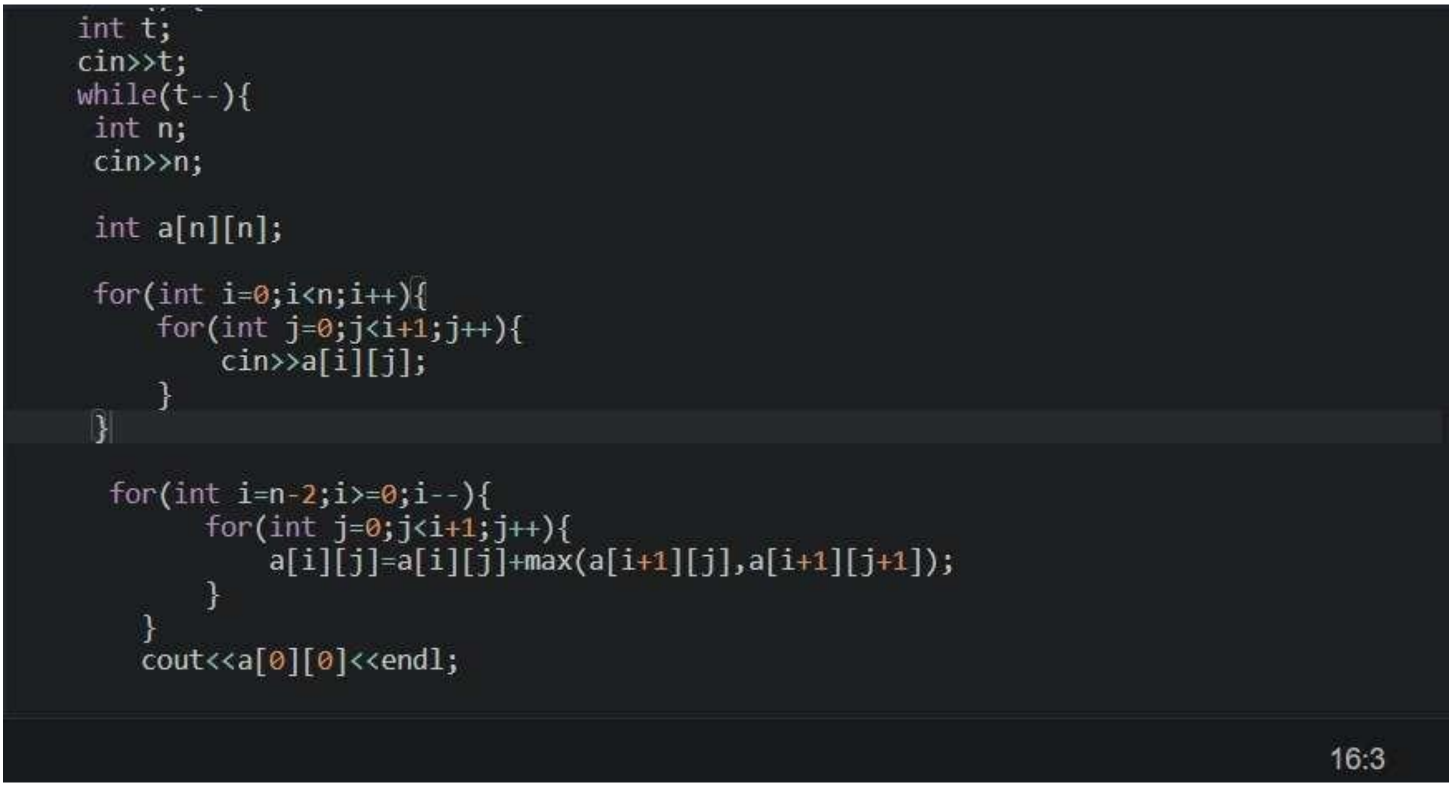


Output

```
1 58
```

4. Sums in a triangle

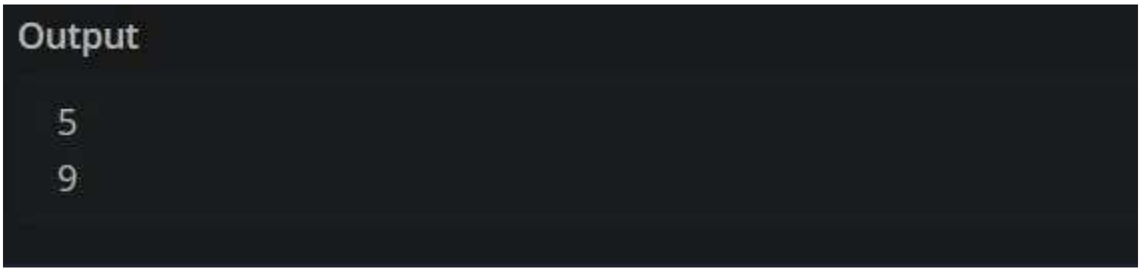
Program Code:



```
int t;
cin>>t;
while(t--){
 int n;
 cin>>n;

 int a[n][n];

 for(int i=0;i<n;i++){
     for(int j=0;j<i+1;j++){
         cin>>a[i][j];
     }
 }

  for(int i=n-2;i>=0;i--){
        for(int j=0;j<i+1;j++){
            a[i][j]=a[i][j]+max(a[i+1][j],a[i+1][j+1]);
        }
    }
    cout<<a[0][0]<<endl;
```

16:3

<u>Output:</u>

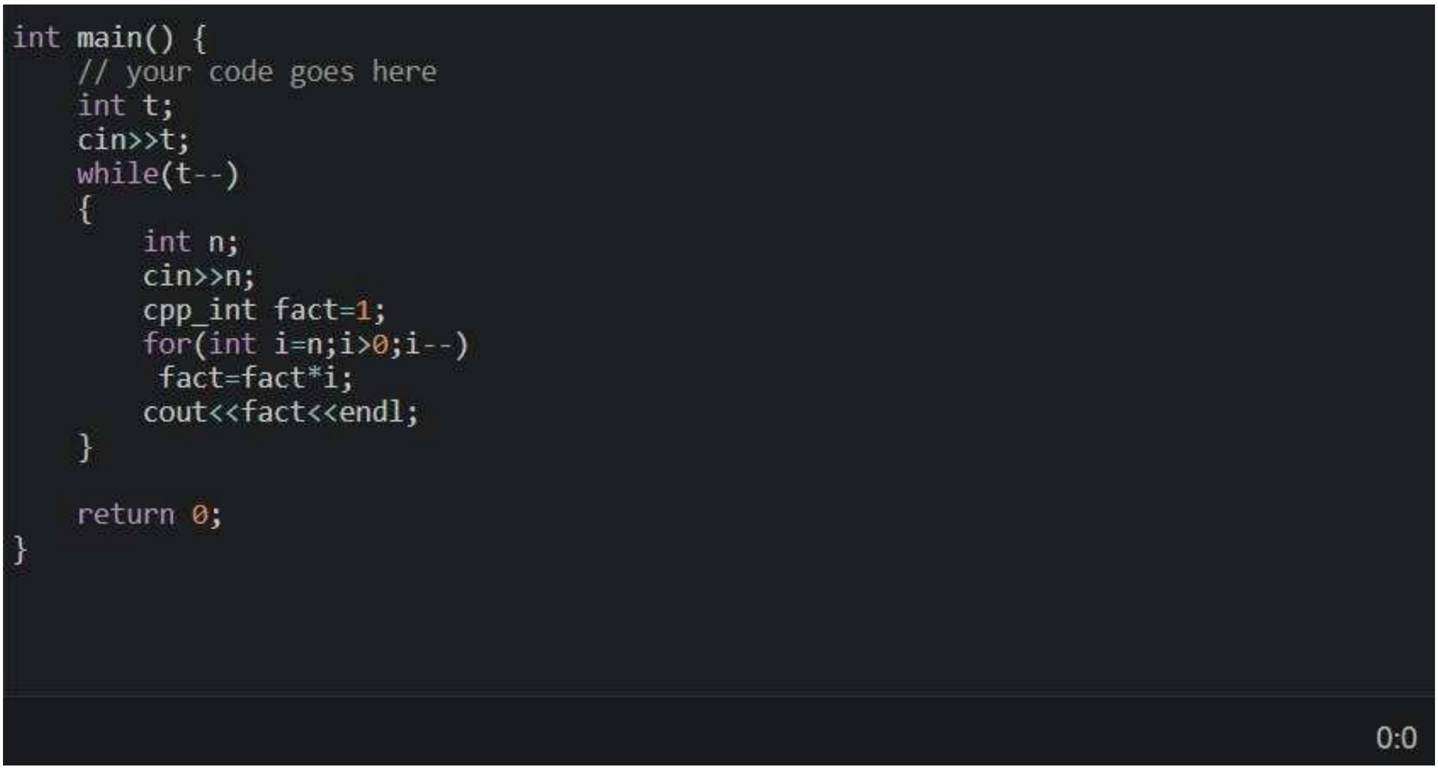


Output

```
5
9
```

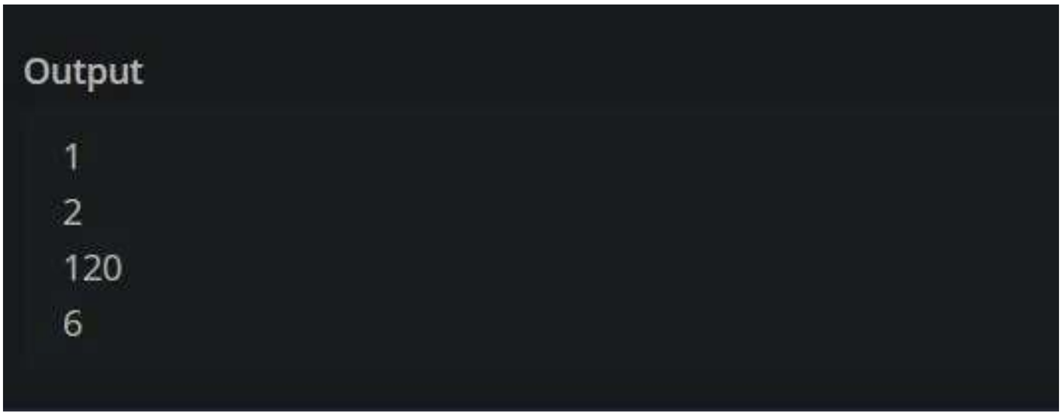
5.Small Factorials

Program Code:



```
int main() {
    // your code goes here
    int t;
    cin>>t;
    while(t--)
    {
        int n;
        cin>>n;
        cpp_int fact=1;
        for(int i=n;i>0;i--)
         fact=fact*i;
        cout<<fact<<endl;
    }

    return 0;
}
```

Output



```
1
2
120
6
```