



# Programming & Problem Solving

**Code: CSE122**

**Lecturer: Rubel Sheikh**



## Team Members:

1. Jobaydul Islam » 222-15-6116

2. Tuhin Sheikh » 222-15-6288

3. Swadhin Biswas » 222-15-6259

4. Md Naimul Islam » 222-15-6299

5. Ismail Gazi » 222-15-6246

6. Fahim Abdullah » 222-15-6163

# :::::Problem:1:::::


- **A particle has initial velocity and constant acceleration. If its velocity after certain time is  $v$  then what will its displacement be in twice of that time?**

## Input

The input will contain two integers in each line. Each line makes one set of input. These two integers denote the value of  $v$  ( $-100 \leq v \leq 100$ ) and  $t$  ( $0 \leq t \leq 200$ ) (  $t$  means at the time the particle gains that velocity). The end of the input is determined by EOF.

## Output

For each line of input print a single integer in one line denoting the displacement in double of that time.



Input Sample	Output Sample
0 0	0
5 12	120

# Algorithm:

Step-1: Start

Step-2: Declare 3 int variable v,t,s then

Step-3: input v. if there is not !=EOF, input t.

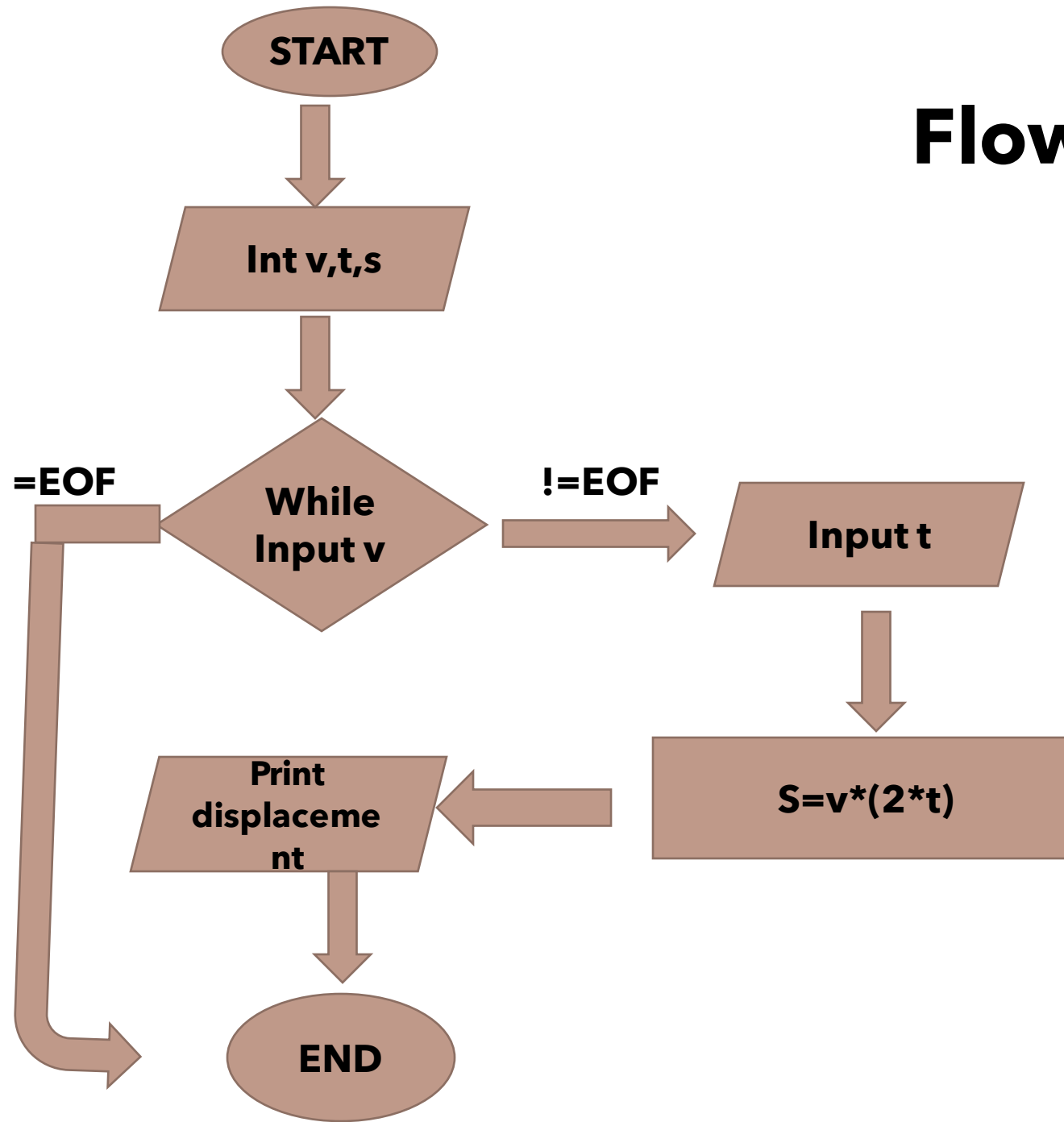
Step-4: calculate value of displacement  $s=v*(2*t)$ ;

Step-5: print displacement .

Step-6:End.



# Flowchart



## Solution code

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

```
int main(){  
int v, t,s;
```

```
while(scanf("%d", &v) != EOF){  
scanf("%d", &t);
```

```
s=v*2*t;
```

```
printf("%d\n", s);  
}  
return 0;  
}
```

## OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
```

```
5 12  
120  
□
```

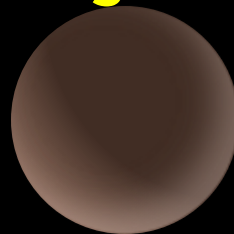
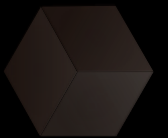
# :::::Problem:::::

- PPS Tracker: 216

- Jumping Frog

- In each stage of the Jumping Frog game you must safely get your amphibian through a sequence of pipes of different heights to the rightmost pipe. Nevertheless the frog just survives if the height difference of consecutive pipes is at most the frog jump height. If the next pipe height is too high, the frog hits the pipe and fall. If the next pipe height is too low, the frog does not survive the fall. The frog always starts on the top of the leftmost pipe.

- In this game the distance of pipes is irrelevant, which means that the frog always can reach the next pipe with a jump. You must write a program that, given the pipe heights and the frog jump height, show if the game stage can be beaten or not.





**Input**

The input is given in two lines. The first one has two positive integer numbers P and N, the frog jump height and the number of pipes ( $1 \leq P \leq 5$  and  $2 \leq N \leq 100$ ). The second line has N positive integer numbers that indicate the pipes heights ordered from left to right. There are no height greater than 10.

**Output**

The output is given in a single line. If the frog can reach the rightmost pipe write "YOU WIN". If the frog fails, write "GAME OVER".

INPUT	OUTPUT
5 10 1 3 6 9 7 2 4 5 8 3	YOU WIN
1 2 2 2	YOU WIN
1 2 1 3	GAME OVER

# Algorithm:

**Step-1: Start**

**Step-2: Take input value of height, num, count=0, step;**

**Step-3: Input array and store num value as jump array input;**

**Step-4: Read the array value by running a loop;**

**(i) Run another for loop value to compare the**

**(ii) if  $\text{jump}[i] > \text{jump}[i-1]$  is true store  $\text{jump}[i] - \text{jump}[i-1]$  in step**

**(iii) else store  $\text{jump}[i-1] - \text{jump}[i-1]$  in step**

**(iv) if step value equal or less than height then increase count value**

**1;**

**Step-5: Now outside the loop check if step is equal num-1 then**

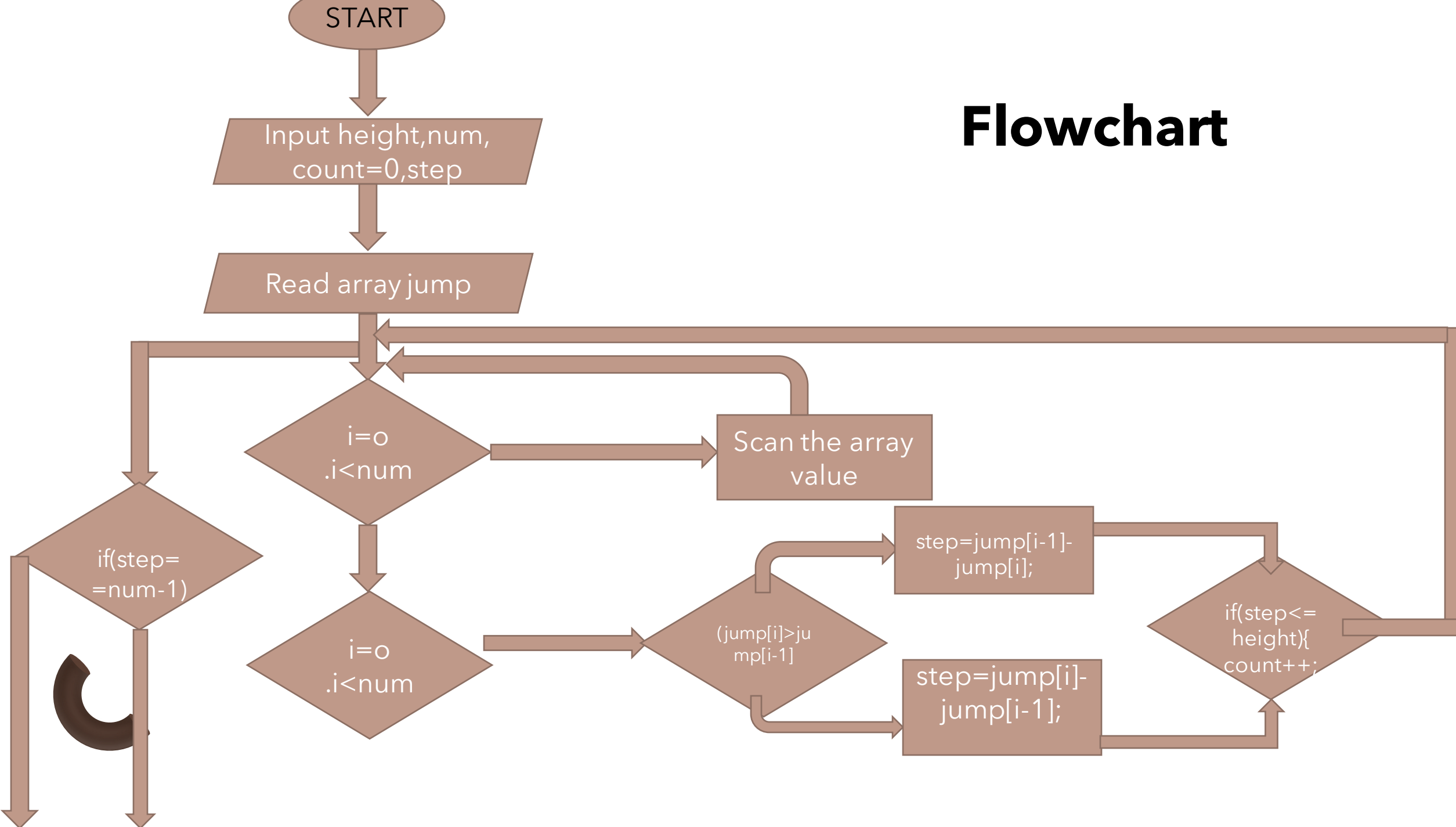
**Step-6: print "YOU WIN"**

**Step-7: Otherwise Print "GAME OVER".**

**Step-8:End.**



# Flowchart





## SOLUTION CODE

```
#include<stdio.h>
int main()
{
    int height,num,count=0,step;//inputing total height,jump number of frog
    scanf("%d %d",&height,&num);
    int jump[num];
    for(int i=0;i<num;i++)//for loop inputing jump number from user
    {
        scanf("%d",&jump[i]);

        for(int i=0;i<num;i++)//for loop for calculate diffrent jump number from user .
        {
            if(jump[i]>jump[i-1])//compare elements index of jump number
            {
                step=jump[i]-jump[i-1];
            }
            else{
                step=jump[i-1]-jump[i];
            }
            if(step<=height){
                count++;
            } //counting all true jump number
        }
    }
}
```

```
}  
}  
  
if(step==num-1)//if there is n time jump we can  
compare each element n-1 time  
{  
printf ("YOU WIN\n");  
}  
else{  
printf ("GAME OVER\n");  
}  
  
return 0;  
  
}
```



## OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
5 7  
2 3 4 5 6 1 3  
GAME OVER  
  
[~/venv]  
swadhin → master
```

*Thank You!*