1. Write a C program to check whether a given number (N) is a perfect number or not?  
   [Perfect Number - A perfect number is a positive integer number which is equals to the sum of its proper positive divisors. For example 6 is a perfect number because its proper divisors are 1, 2, 3 and it’s sum is equals to 6.]

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 8000 | 8000 is not a perfect number. |
| Test Case 2 | 8128 | 8128 is a perfect number. |
| Test Case 3 | 6 | 6 is a perfect number. |
| Test Case 4 | 87 | 87 is not a perfect number. |

#include <stdio.h>

int main()

{

int N;

scanf("%d",&N); /\* An integer number taken as input from test cases \*/

/\*Complete the program by writing the rest of the code in the space provided.

Please copy and paste the printf statement given below wherever required

printf("\n%d is a perfect number.",N);

printf("\n%d is not a perfect number.",N);

\*/

int i, sum=0;

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

{

if(N%i==0)

sum+=i;

}

if(sum==N)

printf("\n%d is a perfect number.",N);

else

printf("\n%d is not a perfect number.",N);

}

1. Write a C program to count total number of digits of an Integer number (N).

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 30001 | The number 30001 contains 5 digits. |
| Test Case 2 | 934521 | The number 934521 contains 6 digits. |
| Test Case 3 | 3456 | The number 3456 contains 4 digits. |
| Test Case 4 | 570 | The number 570 contains 3 digits |

#include <stdio.h>

int main()

{

int N;

scanf("%d",&N); /\*The number is accepted from the test case data\*/

/\* Complete the rest of the code. Please use the printf statements as below

by just changing the variables used in your program

printf("The number %d contains %d digits.",N,count);

\*/

int temp, count;

count=0;

temp=N;

while(temp>0)

{

count++;

temp/=10;

}

printf("The number %d contains %d digits.",N,count);

}

1. Write a C program to check whether the given number(N) can be expressed as Power of Two (2) or not.  
   For example 8 can be expressed as 2^3.

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 6572 | 6572 cannot be expressed as power of 2. |
| Test Case 2 | 1024 | 1024 is a number that can be expressed as power of 2. |
| Test Case 3 | 8 | 8 is a number that can be expressed as power of 2. |
| Test Case 4 | 46 | 46 cannot be expressed as power of 2. |

#include <stdio.h>

int main()

{

int N;

scanf("%d",&N); /\* The value of N is taken from the test case data \*/

/\* Complete the code.

Use the printf statements as below

printf("%d is a number that can be expressed as power of 2.",N);

printf("%d cannot be expressed as power of 2.",N);

\*/

int temp, flag;

temp=N;

flag=0;

while(temp!=1)

{

if(temp%2!=0){

flag=1;

break;

}

temp=temp/2;

}

if(flag==0)

printf("%d is a number that can be expressed as power of 2.",N);

else

printf("%d cannot be expressed as power of 2.",N);

}

1. Write a C program to print the following Pyramid pattern upto Nth row. Where N (number of rows to be printed) is taken as input. For example when the value of N is 5 the pyramid will be printed as follows  
   \*\*\*\*\*  
   \*\*\*\*  
   \*\*\*  
   \*\*  
   \*

#include<stdio.h>

int main()

{

int N;

scanf("%d", &N); /\*The value of N is taken as input from the test case \*/

int i,j;

for(i=N; i>0; i--)

{

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

{

printf("\*");

}

printf("\n");

}

}