**Python program to find ASCII value of a character**

ASCII Value

ASCII (**American Standard Code for Information Interchange**) is the character encoding standard of electronic communication. ASCII codes represent text within computers, telecommunications equipment, and other devices.

Logic

To find the ASCII value of a character we have to use the built-in *ord()* function that returns the Unicode Code Point for the given character.

Program

# Take a character from user

ch = input("Enter any character : ")

# Print the ASCII value

print("\nASCII value of " + ch + " is :", ord(ch))

Output

Enter any character : a  
ASCII value of a is 97

**Python calculate year week and days from given total days**

In this Python program, we are going to see how we can calculate Year, Week and Days from given total days. We all know that 1 year = 365 (ignoring leap year) and 1 week = 7 days on this basis we can determine the year, week and days from given total no of days.

Logic

If we divide the total number of days by 365 it will return a Year. But to calculate week and days logic is a little different. First, we need the remainder value of (Total days % 365) from this value we can determine total no of weeks and days. To calculate total no of weeks simply divide the remainder value by 7 and to calculate total no of days mode (%) it by 7.

Program

# Take input from user

days = int(input("Enter total number of days : "))

# Calculate year, week and days

year = int(days / 365)

temp = int(days % 365)

week = int(temp / 7)

days = int(temp % 7)

# Print the result

print("\nYears :", year)

print("Week :", week)

print("Days :", days)

Output

Enter total number of days : 1212  
Years : 3  
Weeks : 16  
Days : 5

Explanation

**Calculating year –**Divide the total no. of given days with 365.

year = int(days / 365)

**Calculating week**– Mod the total given days with 365 and divide it with 7 (no of weeks).

temp = int(days % 365)

week = temp / 7

**Calculating days** – Mod the total user given days with 365 and again mod the result with 7.

temp = int(days % 365)

days = temp % 7

## Python check whether a character is alphabet or not

In this program, we are going to see whether a user given character is an alphabet or not. To check character is an alphabet or not python has built-in function isalpha*()*.

Program

# Take character input from user

ch = input("Enter any character : ")

# Check for alphabet or not

if ch[0].isalpha() :

print("\n" + ch[0], "is A ALPHABET.")

else :

print("\n" + ch[0], "is NOT A ALPHABET.")

Output

Enter any character : x  
x IS A ALPHABET.

**Python check whether a character is alphabet, digit or special character**

This program is much similar to the previous one but here we are checking whether the given character is an alphabet, digit or a special character.

Function isalpha()

The function isalpha*()* is used to check whether the character is an alphabet or not.

Function isdigit()

The function isdigit*()*is used to check whether the character is a digit or not.

# Take character input from user

ch = input("Enter any character : ")

# Check for alphabet and digit.

if ch[0].isalpha() :

print("\n" + ch[0], "is A ALPHABET.")

elif ch[0].isdigit() :

print("\n" + ch[0], "is A DIGIT.")

else :

print("\n" + ch[0], "is A SPECIAL CHARACTER.")

Output

Enter any character : @   
@ is A SPECIAL CHARACTER.

**Python find largest number among three number using if statement**

In this program, we are going to see how to find the largest number among three numbers using if statement.

Logic

Here we have to compare each number with another two numbers, if it is greater than the both then simply print it.

Let’s say A = 11, B = 22 and C = 15

Then steps would be

1. if **A > B** and **A > C** that means A is the largest number.
2. if **B > A** and **B > C** that means B is the largest number.
3. Similarly if **C > A** and **C > B** that means C is the largest number.

Program

# Take 3 numbers from user

num1 = int(input("Enter 1st number : "))

num2 = int(input("\nEnter 2nd number : "))

num3 = int(input("\nEnter 3rd number : "))

# Check for number 1

if num1 >= num2 and num1 >= num3 :

print("\n" + str(num1), "is largest number")

# Check for number 2

if num2 >= num1 and num2 >= num3 :

print("\n" + str(num2), "is largest number")

# Check for number 3

if num3 >= num1 and num3 >= num2 :

print("\n" + str(num3), "is largest number")

#### Output

Enter 1st number : 10  
Enter 2nd number : 5  
Enter 3rd number : 11  
11 is largest number

**Python find largest number among three number using nested if else statement**

In this program, we are going to find the largest number among three numbers, similar to the previous one, but it is nested if-else version.

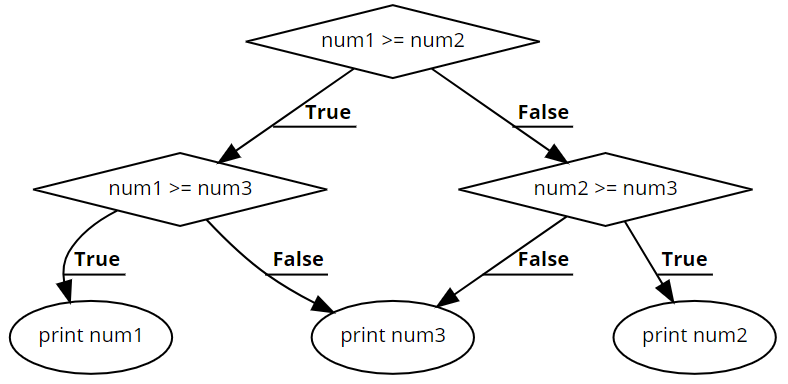
Logic

Let three variables be: A = 400, B = 200 and C = 300

The logic goes like this:

1. if **A >= B**then check for if **A >= C**, then **print A** else **print C.**
2. else part: if **B >= C**then **print B**else **print C.**

See the below-shown flow diagram to better understand.



# Take 3 numbers from user

num1 = int(input("Enter 1st number : "))

num2 = int(input("\nEnter 2nd number : "))

num3 = int(input("\nEnter 3rd number : "))

# Find largest number

if num1 >= num2 :

if num1 >= num3 :

print("\n" + str(num1), "is largest number")

else :

print("\n" + str(num3), "is largest number")

else :

if num2 >= num3 :

print("\n" + str(num2), " is largest number")

else :

print("\n" + str(num3), " is largest number")

#### Output

Enter 1st number : 100  
Enter 2nd number : 10  
Enter 3rd number : 99  
100 is largest number

**Python program to check whether a year is leap year or not**

Leap year

A leap year is a calendar year that includes an additional day to synchronize the calendar year with the astronomical or seasonal year. – [Wikipedia](https://en.wikipedia.org/wiki/Leap_year)

Logic

The Logic to check this is quite simple. We only need to check if the given year is multiple of 4 or 400, but it should not be multiple of 100.

Program

# Take year from user

year = int(input("Enter a year : "))

# Check for leap year

if year % 4 == 0 :

if (year % 100 == 0) :

if (year % 400 == 0) :

print("\n" + str(year), "is a leap year.")

else :

print("\n" + str(year), "is not a leap year.")

else :

print("\n" + str(year), "is a leap year.")

else :

print("\n" + str(year), "is not a leap year.")

#### Output

Enter a year : 2012  
2012 is a leap year.

## Python check whether a character is upper or lowercase alphabet

In this program, we are going to determine if the given character is Uppercase or Lowercase alphabet using python built-in function isupper() and islower().

### **Function isupper()**

The isupper() function can be used to check if the character is Uppercase or not. It will return true if the character is Uppercase character.

### **Function islower()**

The islower() function is opposite to isupper(), it checks if the character is Lowercase character or not.

#### Program

# Take input from user

ch = input("Enter any character : ")[0]

# Check for uppercase, lowercase

if ch.isupper() :

print("\n" + ch, "is UPPERCASE alphabet.")

elif ch.islower() :

print("\n" + ch, "is LOWERCASE alphabet.")

else :

print("\n" + ch, "is not an alphabet.")

#### Output

Enter any caracter : k  
k is LOWERCASE alphabet.

## Python check whether a character is vowel or consonant

There are five vowel characters {**a**, **e**, **i**, **o**, **u**}. If the user given character input is one of them that means it is a vowel otherwise it is a consonant.

### **Logic**

After taking input from the user, we have to check whether the given character is one of character available in the character sequence “aeiou” or not, using python **in** keyword. But before checking first we need to convert it to any one case either lower / upper so that the program can work with both the cases (Uppercase & Lowercase) characters.

### **Function lower()**

Here the function lower() converts the character to lowercase type.

### **Keyword in**

The **in**keyword can be used to check if the value is present in sequence (aeiou) or not.

### **Program**

# Take input from user

ch = input("Enter a alphabet : ")[0]

# if given character is Lower case Vowel or Upper case Vowel

# then print vowel otherwise consonant

if ch.lower() in "aeiou":

print("\n" + ch, "is a vowel.")

else :

print("\n" + ch, "is a consonant.")

#### Output

Enter a alphabet : A  
A is a vowel.

## Python print day name of week from number

In this program, we are going to print day name based on week no. Like – If the user enters 1 that means it is Monday.

Logic

After taking input (Week no) from the user, we have to compare that number with 1 to 7 or we can say comparing that number with a day of a week. So the logic goes like – if the number is equal to 1 then it is Monday, if the number is 2 then it is Tuesday etc… Like that we have to compare with each day of the week.

Program

# Taken day number from user

weekday = int(input("Enter weekday day number (1-7) : "))

if weekday == 1 :

print("\nMonday");

elif weekday == 2 :

print("\nTuesday")

elif(weekday == 3) :

print("\nWednesday")

elif(weekday == 4) :

print("\nThursday")

elif(weekday == 5) :

print("\nFriday")

elif(weekday == 6) :

print("\nSaturday")

elif (weekday == 7) :

print("\nSunday")

else :

print("\nPlease enter weekday number between 1-7.")

#### Output

Enter weekday day number (1-7) : 4  
Thursday

LOOP

## **Python program to find all factors of a number**

Factor

A factor is an integer that can be divided evenly into another number or in other words factors of a number are numbers that multiply to form a product.

Logic

To print all the factors of a particular number we have to iterate through all the smaller numbers up to the given number. If the user given number is completely divisible by any number then it is a factor of that number.

Program

# Taking input from user

num = int(input("Enter any number : "))

print("\nAll factors of", num, "are : ")

for i in range(1, num + 1):

# Completely divisible or not.

if(num % i == 0):

print(i, end = " ")

#### Output

Enter any number : 50  
All factors of 50 are :  
1 2 5 10 25 50

## Python program to print alphabets from a to z

In this program, we are going to see how we can print alphabets in Java. The program is pretty simple and straight forward. Here ASCII value of the characters comes into the picture, we use the ASCII value of starting and end character to run the loop.

See also: [Find ASCII value of a character](https://studyfied.com/program/python-basic/find-ascii-value-of-a-character/)

### **Logic**

ASCII value of small letter **“a”**is 97 and for **“z”**it is 122, so we run the loop from 97 to 122 and then inside the loop we have to print the current character.

### **Program**

# Printing a - z

print("Alphabets from a - z are : ")

# a = 97 and z = 122

for alpha in range(97, 123):

print(chr(alpha), end=" ")

#### Output

Alphabets from a - z are:  
a b c d e f g h i j k l m n o p q r s t u v w x y z

## Python program to check whether a given number is perfect number or not

### **Perfect number**

A perfect number is a positive integer that is equal to the sum of its proper divisors. The smallest perfect number is 6, which is the sum of 1, 2, and 3.

### **Logic**

To check if the number is perfect or not we have to run one loop from 1 to N and sum all the numbers between 1 to N, if the sum is equal to N then it is a perfect number.

### **Program**

# Take input from user

num = int(input("Enter any number : "))

sum = 0

# Calculate sum of all proper divisors

for i in range(1, num):

if num % i == 0:

sum += i

# Empty print statement for new line

print()

# Check whether the sum of divisors is equal to num

if sum == num:

print(num, "is PERFECT NUMBER")

else:

print(num, "is NOT PERFECT NUMBER")

#### Output

Enter any number : 5  
5 is NOT PERFECT NUMBER

**Python program to check whether a number is prime number or not**

Prime number

A prime number is an integer which is greater than 1 whose only factors are 1 and itself. A factor is an integer that can be divided evenly into another number.

Logic

The logic to check a number is prime or not is really simple. We only need to check if the given number is completely divisible by any other smaller number or not, but not by 1 and itself.

After taking input from the user, we have to run one loop up to half of the given number. Now the reason for running one loop up to half of the given number is because we know, a number can only be divisible by a number which is lesser than its half.

Inside the loop we have to check if the given number **“num”** is completely divisible by current number **“i”**or not, If yes then set the value of the variable *i* to *num*and terminate the loop here, if no then continue the loop.

Once the loop is completed or terminated control comes out of the loop, here we have to check if the value of the variable *i* is equivalent to the given number or not. If yes then it is not a Prime number otherwise it is a prime number.

# Take input from user

num = int(input("Enter any number : "))

i = 1

# Check for prime

for i in range(2, num):

if (int(num % i) == 0):

i = num

break;

if i is num:

print("\n" + str(num), "is not a prime number.")

else:

print("\n" + str(num), "is a prime number.")

#### Output

Enter any number : 7  
7 is a prime number.

**Python program to count number of digits in a given integer**

Count number of digits in a given integer

To count the number of digits in a number we have to divide that number by 10 until it becomes 0 or less than 0. If you divide any number by 10 and store the result in an integer then it strips the last digit. So we have to apply the same logic here also.

Logic

After taking the input from the user, run one while loop that runs until the user given number becomes 0. Inside the loop for each iteration increase the counter variable by one and divide the user given number by 10 and store it in the same variable again.

Notice here we are using **/=**operator that performs division and stores the result in the same variable again.

Once the loop is over we have to print counter variable **“count”**containing the total number of digits.

Program

# Take input from user

num = int(input("Enter any number : "))

# Store to temporary variable.

temp = num

count = 0

while (temp != 0):

# Increment counter

count += 1

# Remove last digit of 'temp'

temp = int(temp / 10);

print("\nTotal digits in", num, ":", count)

#### Output

Enter any number : 123456  
Total digits in 123456 : 6

**Python program to find product of digits in a number**

Product of digits in a number

This program is closely similar to this one: [Count number of digits in a given integer](https://studyfied.com/program/python-basic/count-number-of-digits-in-a-given-integer/). The only difference here is instead of counting the total number of digits we are multiplying each digit by another one until the user given number becomes 0 or less than 0.

Logic

First of all, we are declaring one variable *product*with value 1, we are going to use this variable to store the product of all digits, now after taking input from the user, inside the while loop, we have to extract each digit by performing modulo operation.

The modulo operation returns the remainder after dividing a number by another number. If we perform modulo operation with 10 on any number it will return the last most digit, we have to multiply and store the result into the variable *product.*

After that remove the last most digit from the number and perform the same again until the number becomes 0 or less than 0. Once the loop is completed simple print the variable *product* containing the product of all digit.

Program

# Take input from user

num = int(input("Enter any number : "))

temp = num

product = 1;

while(temp != 0):

product = product \* (temp % 10);

# Remove last digit from temp.

temp = int(temp / 10)

print("\nProduct of all digits in", num, ":", product)

#### Output

Enter any number : 123456  
Product of all digits in 123456 : 720

**Python program to find HCF of two user given numbers**

HCF

The greatest common divisor of two or more numerical values is called the highest common factor (HCF). In other words, the largest number that can completely divide two or more numbers is the highest common factor.

Logic

Let declare one temporary variable *hcf* by 1, we are going to use this variable to store HCF.

Here we are calculating HCF of two user given numbers, after taking both the numbers (num1, num2) from the user, we have to find out which number is smaller, we store the minimum one in the variable *min*.

Now we start one loop from 1 to *min*(Minimum between both numbers), and inside the loop, we have to check if both the numbers are completely divisible by current number **“i”**or not. If yes then we update temporary variable *hcf*by current number **“i”**, otherwise continue with the loop.

Once the loop is over simply print the variable *hcf*containing the highest common factor of num1 and num2.

Program

# Take input from user

num1 = int(input("Enter 1st number : "))

num2 = int(input("Enter 2nd number : "))

# Find minimum between two numbers

min = num1

if num2 < num1:

min = num2

for i in range(1, min + 1):

if((num1 % i) == 0 and (num2 % i) == 0):

hcf = i

print("\nHCF of", num1, "and", num2, ":", hcf)

#### Output

Enter two numbers : 12 30  
HCF of 12 and 30 : 6

**Python program to find LCM of two user given numbers**

LCM

The least / lowest common multiple (LCM) of two or more than two numbers is the smallest number (not zero) which is a multiple of all of the numbers.

Logic

Let declare one temporary variable *lcm* by 1, we are going to use this variable to store LCM.

Here we are calculating LCM of two user given numbers, after taking both the numbers (num1, num2) from the user, we have to find out the maximum number, we store the maximum one in variable *max* and also assign the value of max to the variable *i.*

Now we start one infinite loop *while(1),*inside the loop we have to check if both the numbers can completely divide *max*value. If yes then we terminate the loop with *break*statement and assign current value of **“i”**to *lcm,*otherwise, continue the loop.

Once the loop is over, simply print the variable *lcm*containing the lowest common multiple of num1 and num2.

Program

# Take input from user

num1 = int(input("Enter 1st number : "))

num2 = int(input("Enter 2nd number : "))

# Find the max number

max\_num = num2

if (num1 > num2):

max\_num = num1

i = max\_num;

lcm = 1;

while(True):

if((i % num1) == 0 and (i % num2) == 0):

lcm = i;

break;

i += max\_num;

print("\nLCM of", num1, "and", num2, ":", lcm)

#### Output

Enter any two number : 6 24  
LCM of 6 and 24 : 24