

**“AZƏRBAYCAN HAVA YOLLARI” CJSC NATIONAL AVIATION ACADEMY**

**Individual Work №: 2**

**Topic: Space complexity in Python**

**Subject: OP Teacher: Mamed Shahmaliyev**

**Group: 1459i Student: Ahmad Mammadov**

**Date: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Baku 2022**

What is Space Complexity?

Space complexity is the **total amount of memory space used by an algorithm/program including the space of input values for execution**. So to find space-complexity, it is enough to calculate the space occupied by the variables used in an algorithm/program.

But often, people confuse Space-complexity with Auxiliary space. Auxiliary space is just a temporary or extra space and it is not the same as space-complexity. In simpler terms,

**Space Complexity = Auxiliary space + Space use by input values**

**﻿Important Note:**The best algorithm/program should have the lease space-complexity. The lesser the space used, the faster it executes.

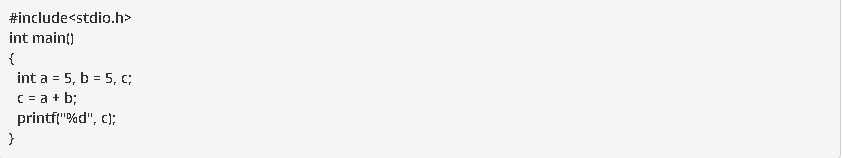
## **Why do you need to calculate space complexity?**

Similar to Time Complexity, Space-complexity also plays a crucial role in determining the efficiency of an algorithm/program. If an algorithm takes up a lot of time, you can still wait, run/execute it to get the desired output. But, **if a program takes up a lot of memory space, the compiler will not let you run it**.

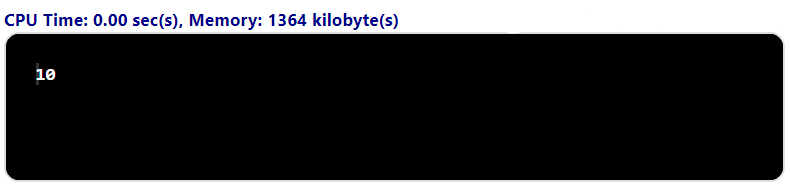
## **How to calculate Space Complexity of an Algorithm?**

Let us understand the Space-Complexity calculation through examples.

### **Example #1**



**Output:**



**Explanation:**Do not misunderstand space-complexity to be 1364 Kilobytes as shown in the output image. The method to calculate the actual space complexity is shown below.

In the above program, 3 integer variables are used. The size of the integer data type is 2 or 4 bytes which depends on the compiler. Now, lets assume the size as 4 bytes. So, the total space occupied by the above-given program is 4 \* 3 = 12 bytes. Since no additional variables are used, no extra space is required.

Hence, **space complexity for the above-given program is O(1), or constant.**

### **Example #2**

#include <stdio.h>

int main()

{

int n, i, sum = 0;

scanf("%d", &n);

int arr[n];

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

{

scanf("%d", &arr[i]);

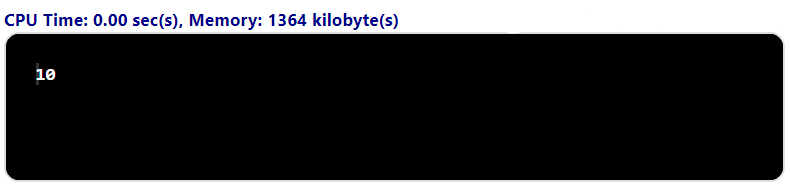
sum = sum + arr[i];

}

printf("%d", sum);

}

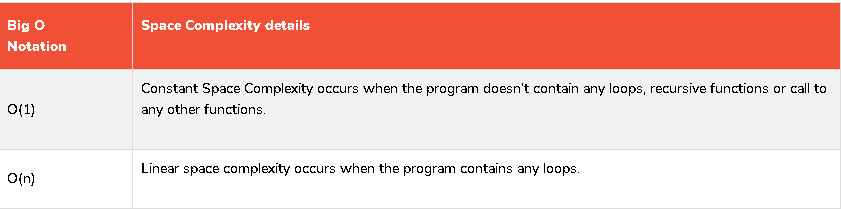
**Output:**



**Explanation:**

In the above-given code, the array consists of n integer elements. So, the space occupied by the array is 4 \* n. Also we have integer variables such as n, i and sum. Assuming 4 bytes for each variable, the total space occupied by the program is 4n + 12 bytes. Since the highest order of n in the equation 4n + 12 is n, so **the space complexity is O(n) or linear.**

## **Summary**



**Space complexity cheat sheet for algorithms**

