



# Perfect Plan B

## Perfect Plan B

Learn, grow and become leaders of tomorrow

# Problem : Python program to interchange first and last elements in a list

Examples:

Input : [12, 35, 9, 56, 24]

Output : [24, 35, 9, 56, 12]

Input : [1, 2, 3]

Output : [3, 2, 1]

Perfect Plan B

# Solution 1 : Python program to interchange first and last elements in a list

```
# Python3 program to swap first  
# and last element of a list
```

```
# Swap function
```

```
def swapList(newList):  
    size = len(newList)
```

```
    # Swapping
```

```
    temp = newList[0]
```

```
    newList[0] = newList[size - 1]
```

```
    newList[size - 1] = temp
```

```
    return newList
```

```
# Driver code
```

```
newList = [12, 35, 9, 56, 24]
```

```
print(swapList(newList))
```

Perfect  Plan B

## Solution 2 : Python program to interchange first and last elements in a list

```
# Python3 program to swap first  
# and last element of a list
```

```
# Swap function
```

```
def swapList(newList):
```

```
    newList[0], newList[-1] = newList[-1], newList[0]
```

```
    return newList
```

```
# Driver code
```

```
newList = [12, 35, 9, 56, 24]
```

```
print(swapList(newList))
```

Perfect Plan B

# Solution 3 : Python program to interchange first and last elements in a list

```
# Python3 program to swap first  
# and last element of a list
```

```
# Swap function  
def swapList(list):
```

```
    # Storing the first and last element  
    # as a pair in a tuple variable get  
    get = list[-1], list[0]
```

```
    # unpacking those elements  
    list[0], list[-1] = get
```

```
    return list
```

```
# Driver code  
newList = [12, 35, 9, 56, 24]  
print(swapList(newList))
```

Perfect Plan B

# Solution 4 : Python program to interchange first and last elements in a list

```
# Python3 program to swap first  
# and last element of a list
```

```
# Swap function  
def swapList(list):
```

```
    first = list.pop(0)  
    last = list.pop(-1)
```

```
    list.insert(0, last)  
    list.append(first)
```

```
    return list
```

```
# Driver code  
newList = [12, 35, 9, 56, 24]
```

```
print(swapList(newList))
```

Perfect  Plan B

# Problem : Python program to check if a string is palindrome or not

Examples:

Input : malayalam  
Output : Yes

Input : geeks  
Output : No

Perfect  Plan B

# Solution 1 : Python program to check if a string is palindrome or not

# function which return reverse of a string

```
def isPalindrome(s):  
    return s == s[::-1]
```

# Driver code

```
s = "malayalam"
```

```
ans = isPalindrome(s)
```

```
if ans:
```

```
    print("Yes")
```

```
else:
```

```
    print("No")
```

Perfect  Plan B



# Solution 2 : Python program to check if a string is palindrome or not

```
# function to check string is  
# palindrome or not  
def isPalindrome(str):
```

```
    # Run loop from 0 to len/2  
    for i in xrange(0, len(str)/2):  
        if str[i] != str[len(str)-i-1]:  
            return False  
    return True
```

```
# main function  
s = "malayalam"  
ans = isPalindrome(s)
```

```
if (ans):  
    print("Yes")  
else:  
    print("No")
```

# Solution 3 : Python program to check if a string is palindrome or not

```
# function to check string is  
# palindrome or not  
def isPalindrome(s):
```

```
    # Using predefined function to  
    # reverse to string print(s)  
    rev = ''.join(reversed(s))
```

```
    # Checking if both string are  
    # equal or not  
    if (s == rev):  
        return True  
    return False
```

```
# main function  
s = "malayalam"  
ans = isPalindrome(s)
```

Problem : Python program to create a list of tuples from given list having number and its cube in each tuple

**Example:**

Input: list = [1, 2, 3]

Output: [(1, 1), (2, 8), (3, 27)]

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6,

216)]

# Solution : Python program to create a list of tuples from given list having number and its cube in each tuple

```
# Python program to create a list of tuples  
# from given list having number and  
# its cube in each tuple
```

```
# creating a list  
list1 = [1, 2, 5, 6]
```

```
# using list comprehension to iterate each  
# values in list and create a tuple as specified  
res = [(val, pow(val, 3)) for val in list1]
```

```
# print the result  
print(res)
```

Perfect  Plan B

# Problem : Python | Print an Inverted Star Pattern

## Examples:

Below is the inverted star pattern of size  
n=5

(Because there are 5 horizontal lines  
or rows consist of stars).

```
*****  
  ****  
   ***  
    **  
     *  
    
```

# Solution : Python | Print an Inverted Star Pattern

```
# python 3 code to print inverted star  
# pattern
```

```
# n is the number of rows in which  
# star is going to be printed.  
n=11
```

```
# i is going to be enabled to  
# range between n-i t 0 with a  
# decrement of 1 with each iteration.  
# and in print function, for each  
iteration,  
# " " is multiplied with n-i and "*" is  
# multiplied with i to create correct  
# space before of the stars.  
for i in range (n, 0, -1):  
    print((n-i) * ' ' + i * '*')
```

Perfect Plan B

# About Piazza

Please watch this video about how to use Piazza:

<https://youtu.be/ndzsh5ShhhA>

Perfect  Plan B

# Social Media Links

**Facebook:** <https://www.facebook.com/P2BGlobal/>

**Twitter:** <https://twitter.com/P2BGlobal>

**Linkedin:** <https://www.linkedin.com/company/p2bglobal>

**Instagram:** <https://www.instagram.com/p2bglobal>

**Quora:** <https://qr.ae/pN25Iq>

**Youtube:** <https://www.youtube.com/c/P2BGlobal>