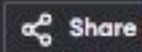


main.c



Run

Output

```
1  #include<stdio.h>
2  #include<string.h>
3  int pal(char *str)
4  {
5      int left=0;
6      int right=strlen(str)-1;
7      while(left<right)
8      {
9          if(str[left]!=str[right])
10         {
11             return 0;
12         }
13         left++;
14         right--;
15     }
16     return 1;
17 }
18 const char* firstpal(char *a[],int b)
19 {
20     for(int i=0;i<b;i++)
21     {
22         if(pal(a[i]))
23         {
24             return a[i];
25         }
26     }
27     return NULL;
28 }
29
30 int main()
31 {
32     char *a[]={"abc","car","racecar","cool","ada"};
33     printf("%s\n",firstpal(a,5));
34     return 0;
```

/tmp/pVFCzH15dU.o

racecar

=== Code Execution Successful ===

main.py



Share

Run

Output

```
1 a=[2,4,5,3,6]
2 b=[1,2,4,5,6]
3 ans1=0
4 ans2=0
5 for i in range(len(a)):
6     if a[i] in b:
7         ans1+=1
8 for j in range(len(b)):
9     if b[j] in a:
10        ans2+=1
11 print("the numbers at indices",i,"are",ans1,ans2)
```

the numbers at indices 4 are 4 4

=== Code Execution Successful ===



main.py



Share

Run

Output

```
1 num=[1,2,1]
2 n=len(num)
3 sum=0
4 distinct=set()
5 for i in range(n):
6     distinct.clear()
7     for j in range(i,n):
8         distinct.add(num[j])
9         sum+=len(distinct)**2
10 print(sum)
```

15

=== Code Execution Successful ===

main.py



Share

Run

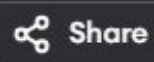
Output

```
1 num=[1,2,3,1,4,2,0,1]
2 k=2
3 n=len(num)
4 c=0
5 for i in range(n):
6     for j in range(i+1,n):
7         if num[i]==num[j] and (i*j)%k==0:
8             c+=1
9 print(c)
```

2

=== Code Execution Successful ===S

main.py



Share

Run

Output

```
1 def maxnumber(numbers):  
2     return max(numbers)  
3 print(maxnumber([1,2,3,4,5]))  
4 print(maxnumber([7,7,7,7,7]))  
5 print(maxnumber([-10,2,3,-4,-5]))
```

5

7

3

=== Code Execution Successful ===

main.py



Share

Run

Output

```
1 def sort_numbers(numbers):
2     sorted_numbers=sorted(numbers)
3     max_element=max(sorted_numbers)
4     return sorted_numbers,max_element
5 numbers=[3,1,4,1,5,9,2,6]
6 sorted_numbers,max_element=sort_numbers(numbers)
7 print("sorted elements:",sorted_numbers)
8 print("max_element",max_element)
```

```
sorted elements: [1, 1, 2, 3, 4, 5, 6, 9]
max_element 9
```

```
=== Code Execution Successful ===
```

main.py



Share

Run

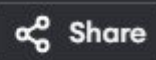
Output

```
1 def unique(given_list):
2     unique_list=[]
3     for a in given_list:
4         if a not in unique_list:
5             unique_list.append(a)
6     return unique_list
7 num=[1,2,1,2,3,4,4,5]
8 print(unique(num))
```

[1, 2, 3, 4, 5]

=== Code Execution Successful ===

main.py



Share

Run

Output

```
1 def bubble(a):
2     n=len(a)
3     for i in range(n):
4         for j in range(0,n-i-1):
5             if a[j]>a[j+1]:
6                 a[j],a[j+1]=a[j+1],a[j]
7     return a
8 arr=[54,12,36,48,98,43]
9 print("the sorted array is",bubble(arr))
```

the sorted array is [12, 36, 43, 48, 54, 98]

=== Code Execution Successful ===

main.py



Share

Run

Output

```
1 def bin_search(a,x):
2     left,right=0,len(a)-1
3     while left<=right:
4         mid=left+(right-left)//2
5         if a[mid]==x:
6             return True
7         elif a[mid]<x:
8             left=mid+1
9         else:
10            right=mid-1
11    return False
12 sortedarray=[1,5,2,4,6,8,9,0]
13 sortedarray.sort()
14 b=4
15 result=bin_search(sortedarray,b)
16 print(f"number {b} is found {result}")
```

number 4 is found True

=== Code Execution Successful ===

main.py



Share

Run

Output

```
1 def quick(nums):
2     if len(nums)<=1:
3         return nums
4     pivot=nums[len(nums)//2]
5     left=[x for x in nums if x<pivot]
6     middle=[x for x in nums if x==pivot]
7     right=[x for x in nums if x>pivot]
8     return quick(left)+middle+quick(right)
9 num=[38,27,43,3,9,82,10]
10 sorted_numbers=quick(num)
11 print(sorted_numbers)
```

[3, 9, 10, 27, 38, 43, 82]

=== Code Execution Successful ===