main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
int pal(char *str)
{
    int left=0;
    int right=strlen(str)-1;
    while(left<right)
    {
        if(str[left]!=str[right])
        {
            return 0;
        }
        left++;
        right--;
    }
    return 1;
}
const char* firstpal(char *a[],int b)
{
    for(int i=0;i<b;i++)
    {
        if(pal(a[i]))
        {
            return a[i];

        }
    }
    return NULL;
}
int main()
{
    char *a[]={"abc","car","racecar","cool","ada"};
    printf("%s\n",firstpal(a,5));
    return 0;
```

Output

```
/tmp/pVFCzH15dU.o
racecar


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

main.py

Share Run

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

Output

```
the numbers at indices 4 are 4 4

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

Programiz

Python Online Compiler

main.py

Share

Run

```python
num=[1,2,1]
n=len(num)
sum=0
distinct=set()
for i in range(n):
    distinct.clear()
    for j in range(i,n):
        distinct.add(num[j])
        sum+=len(distinct)**2
print(sum)
```

Output

```
15

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

main.py

Share Run

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

Output

```
2

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

main.py

Share Run

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

Output

```
5
7
3

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

main.py

Share Run

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

Output

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

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

main.py

Share Run

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

Output

```
[1, 2, 3, 4, 5]

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

main.py

Share Run

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

Output

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

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

main.py

Share Run

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

Output

```
number 4 is found True

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

main.py

Share Run

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

Output

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

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