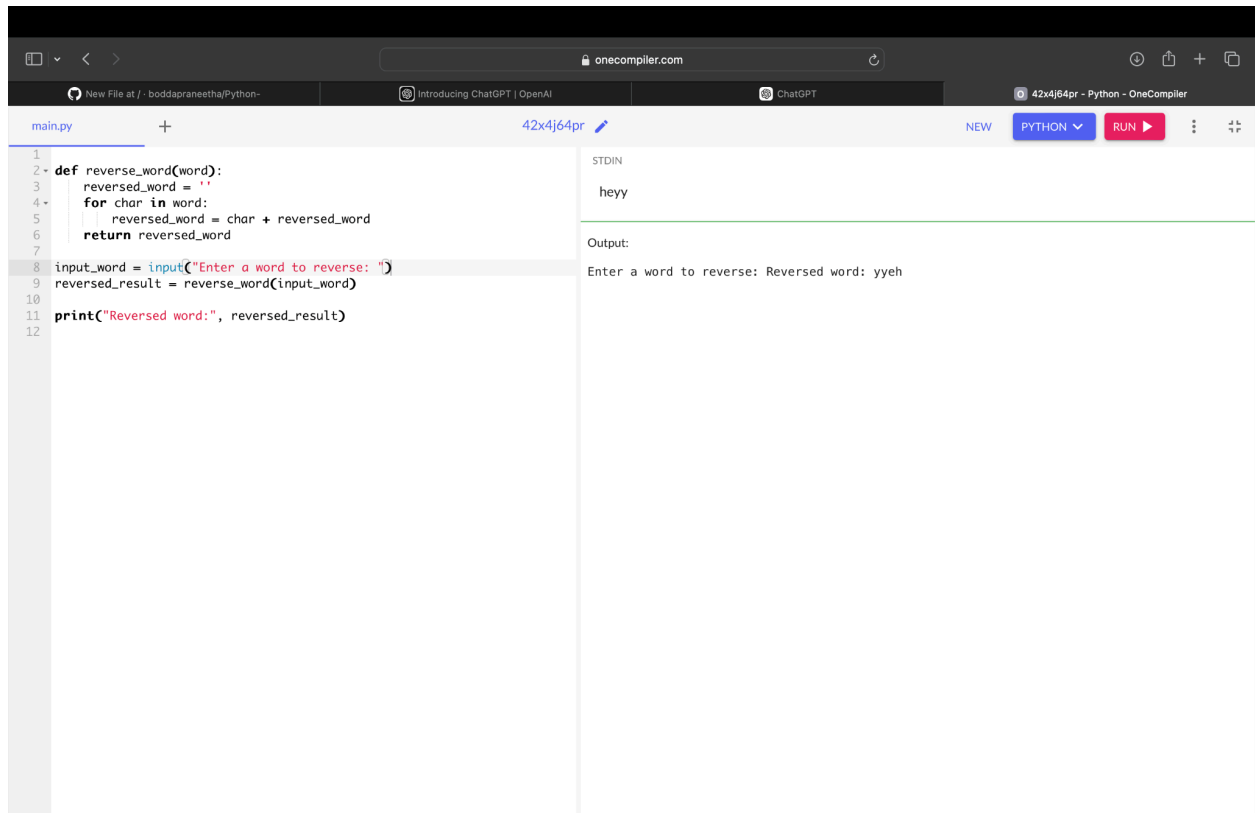


1. Write a python program to reverse a word using loop?



The screenshot shows the OneCompiler web interface. The browser address bar displays 'onecompiler.com'. The editor has a single file named 'main.py' open. The Python code in the editor is as follows:

```
1
2 def reverse_word(word):
3     reversed_word = ''
4     for char in word:
5         reversed_word = char + reversed_word
6     return reversed_word
7
8 input_word = input("Enter a word to reverse: ")
9 reversed_result = reverse_word(input_word)
10
11 print("Reversed word:", reversed_result)
12
```

On the right side of the editor, the 'STDIN' input is 'hey' and the 'Output' section shows the result of running the program: 'Enter a word to reverse: Reversed word: yyeh'.

2. Write a python program to calculate Pow(x,n), Add(x,n), Sub(x,n), Mul(x,n), Div(x,n)? Get the input and choice from the user.

```

10 def calculate_pow(x, n):
11     return x ** n
12
13 def calculate_div(x, n):
14     if n != 0:
15         return x / n
16     else:
17         return "Error: Division by zero is not allowed."
18
19 # Get user input
20 x = float(input("Enter the value of x: "))
21 n = float(input("Enter the value of n: "))
22
23 # Display operation choices
24 print("Choose an operation:")
25 print("1. Power (x^n)")
26 print("2. Addition (x + n)")
27 print("3. Subtraction (x - n)")
28 print("4. Multiplication (x * n)")
29 print("5. Division (x / n)")
30
31 choice = input("Enter the number corresponding to your choice (1-5): ")
32
33 # Perform the chosen operation
34 if choice == '1':
35     result = calculate_pow(x, n)
36     print(f"{x} raised to the power of {n} is {result}.")
37 elif choice == '2':
38     result = calculate_add(x, n)
39     print(f"{x} plus {n} is {result}.")
40 elif choice == '3':
41     result = calculate_sub(x, n)
42     print(f"{x} minus {n} is {result}.")
43 elif choice == '4':
44     result = calculate_mul(x, n)
45     print(f"{x} times {n} is {result}.")
46 elif choice == '5':
47     result = calculate_div(x, n)
48     print(f"{x} divided by {n} is {result}.")
49 else:
50     print("Invalid choice! Please enter a number between 1 and 5.")
51

```

Output:

Enter the value of x: Enter the value of n: Choose an operation:
1. Power (x^n)
2. Addition (x + n)
3. Subtraction (x - n)
4. Multiplication (x * n)
5. Division (x / n)
Enter the number corresponding to your choice (1-5): 2.36 plus 3.56 is 5.92.

3. Write a python program to count all the prime and composite numbers entered by the user.

```

1 def is_prime(num):
2     if num <= 1:
3         return False
4     for i in range(2, int(num**0.5) + 1):
5         if num % i == 0:
6             return False
7     return True
8
9 def main():
10     prime_count = 0
11     composite_count = 0
12
13     numbers = input("Enter numbers separated by spaces: ")
14     number_list = map(int, numbers.split())
15
16     for number in number_list:
17         if is_prime(number):
18             prime_count += 1
19         elif number > 1:
20             composite_count += 1
21
22     print(f"Prime numbers count: {prime_count}")
23     print(f"Composite numbers count: {composite_count}")
24
25 if __name__ == "__main__":
26     main()
27

```

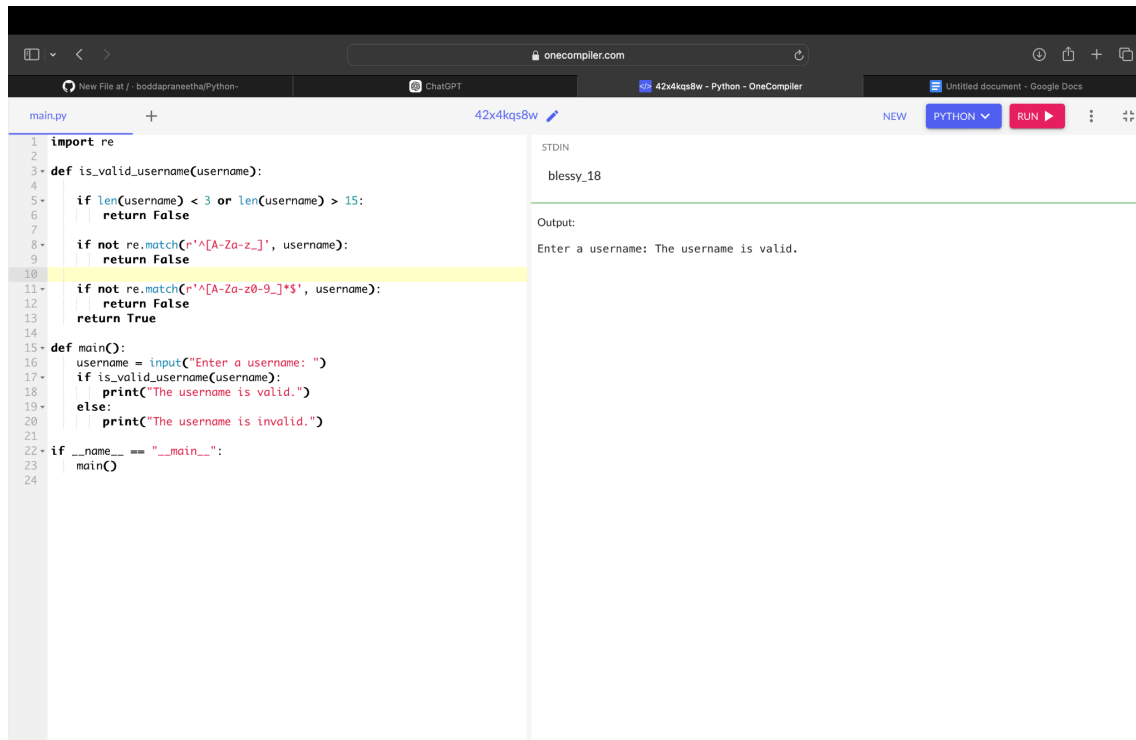
STDIN

5 10 3 5 9 3 11 19 14 15

Output:

Enter numbers separated by spaces: Prime numbers count: 6
Composite numbers count: 4

4. Write a python program to check the entered username is valid or not. Get both the inputs from the user.



```
1 import re
2
3 def is_valid_username(username):
4     if len(username) < 3 or len(username) > 15:
5         return False
6     if not re.match(r'^[A-Za-z_]', username):
7         return False
8     if not re.match(r'^[A-Za-z0-9_]*$', username):
9         return False
10    return True
11
12 def main():
13     username = input("Enter a username: ")
14     if is_valid_username(username):
15         print("The username is valid.")
16     else:
17         print("The username is invalid.")
18
19 if __name__ == "__main__":
20     main()
```

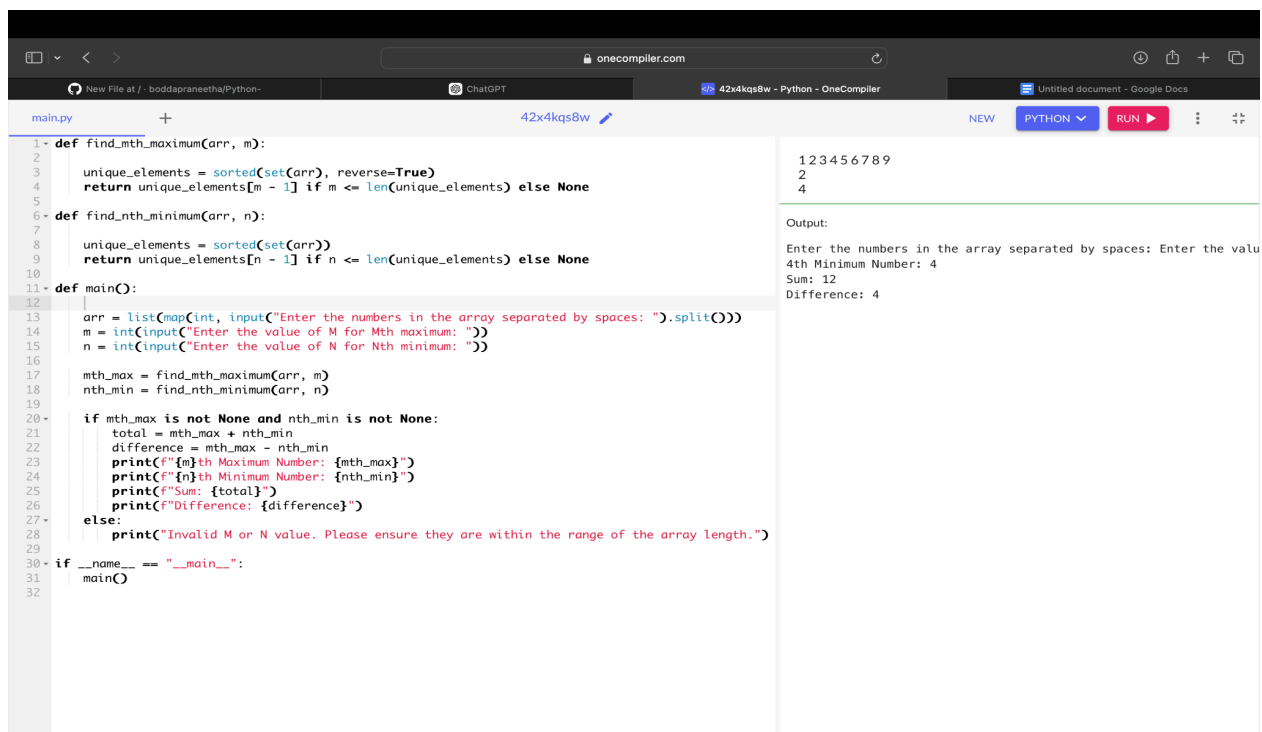
STDIN

blessy_18

Output:

Enter a username: The username is valid.

5. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.



```
1 def find_mth_maximum(arr, m):
2     unique_elements = sorted(set(arr), reverse=True)
3     return unique_elements[m - 1] if m <= len(unique_elements) else None
4
5 def find_nth_minimum(arr, n):
6     unique_elements = sorted(set(arr))
7     return unique_elements[n - 1] if n <= len(unique_elements) else None
8
9 def main():
10    arr = list(map(int, input("Enter the numbers in the array separated by spaces: ").split()))
11    m = int(input("Enter the value of M for Mth maximum: "))
12    n = int(input("Enter the value of N for Nth minimum: "))
13
14    mth_max = find_mth_maximum(arr, m)
15    nth_min = find_nth_minimum(arr, n)
16
17    if mth_max is not None and nth_min is not None:
18        total = mth_max + nth_min
19        difference = mth_max - nth_min
20        print(f"{m}th Maximum Number: {mth_max}")
21        print(f"{n}th Minimum Number: {nth_min}")
22        print(f"Sum: {total}")
23        print(f"Difference: {difference}")
24    else:
25        print("Invalid M or N value. Please ensure they are within the range of the array length.")
26
27 if __name__ == "__main__":
28     main()
```

1 2 3 4 5 6 7 8 9

2

4

Output:

Enter the numbers in the array separated by spaces: Enter the value of M for Mth maximum: 4

4th Minimum Number: 4

Sum: 12

Difference: 4

6. Write a program to reverse a number using loop?(Get the input from user)

The screenshot shows a web browser window with the URL `onecompiler.com`. The browser has several tabs open, including "1 to 10 - Google Docs", "42xa3uys3 - Python - OneCompiler", "Other Practice Programs for Python.docx - Google Docs", and "ChatGPT". The active tab is "42xa3uys3 - Python - OneCompiler". The code editor shows a file named `main.py` with the following Python code:

```
1 number = int(input("Enter a number: "))
2
3 reversed_number = 0
4
5 while number != 0:
6     last_digit = number % 10
7     reversed_number = reversed_number * 10 + last_digit
8     number = number // 10
9
10 print("Reversed number:", reversed_number)
```

The right side of the interface shows the execution results. Under "STDIN", the input `1234` is shown. Under "Output:", the text `Enter a number: Reversed number: 4321` is displayed.

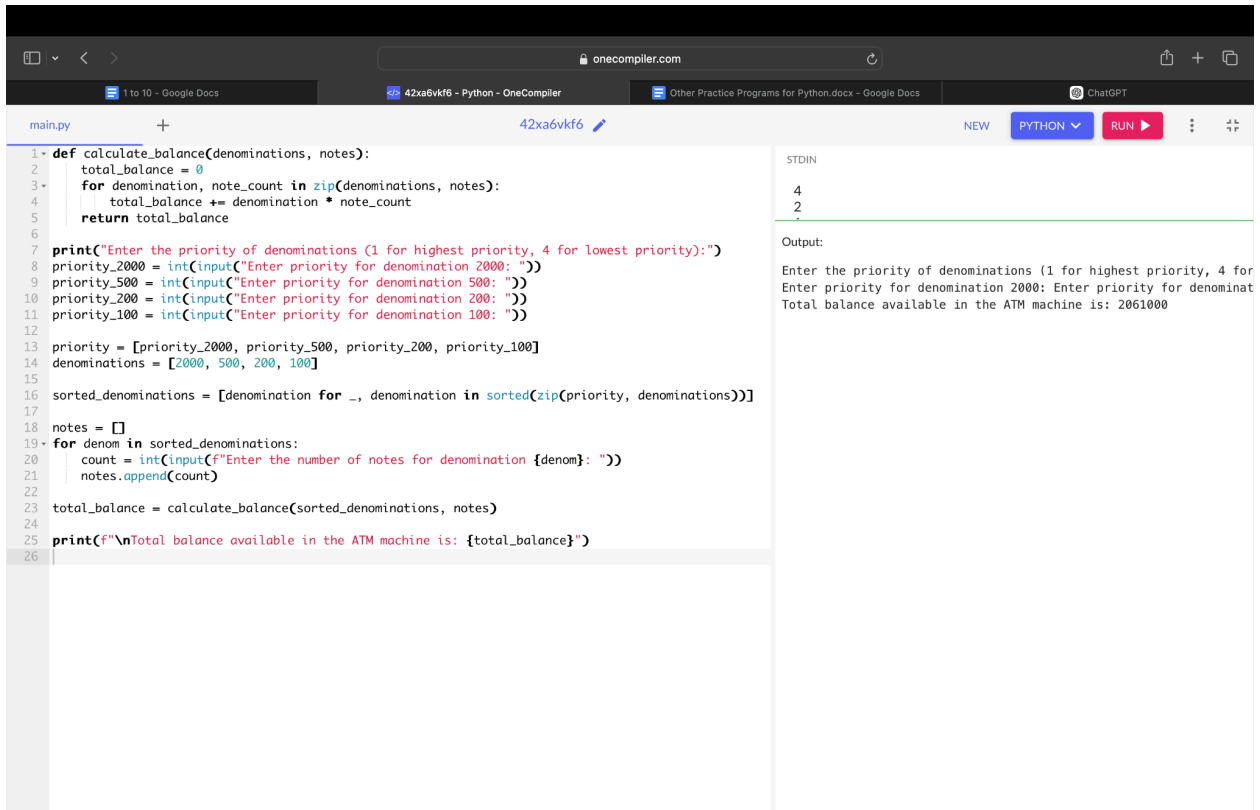
7. Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

The screenshot shows the same web browser window with the URL `onecompiler.com`. The active tab is "42xa3uys3 - Python - OneCompiler". The code editor shows a file named `main.py` with the following Python code:

```
1 age = int(input("Enter your age: "))
2
3 voting_age = 18
4
5 if age >= voting_age:
6     print("You are eligible to vote!")
7 else:
8     years_left = voting_age - age
9     print(f"You are not eligible to vote.")
```

The right side of the interface shows the execution results. Under "STDIN", the input `12` is shown. Under "Output:", the text `Enter your age: You are not eligible to vote.` is displayed.

8. Write a program to print the total amount available in the ATM machine with the conditions applied. Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user



```
1- def calculate_balance(denominations, notes):
2-     total_balance = 0
3-     for denomination, note_count in zip(denominations, notes):
4-         total_balance += denomination * note_count
5-     return total_balance
6-
7- print("Enter the priority of denominations (1 for highest priority, 4 for lowest priority):")
8- priority_2000 = int(input("Enter priority for denomination 2000: "))
9- priority_500 = int(input("Enter priority for denomination 500: "))
10- priority_200 = int(input("Enter priority for denomination 200: "))
11- priority_100 = int(input("Enter priority for denomination 100: "))
12-
13- priority = [priority_2000, priority_500, priority_200, priority_100]
14- denominations = [2000, 500, 200, 100]
15-
16- sorted_denominations = [denomination for _, denomination in sorted(zip(priority, denominations))]
17-
18- notes = []
19- for denom in sorted_denominations:
20-     count = int(input(f"Enter the number of notes for denomination {denom}: "))
21-     notes.append(count)
22-
23- total_balance = calculate_balance(sorted_denominations, notes)
24-
25- print(f"\nTotal balance available in the ATM machine is: {total_balance}")
26-
```

STDIN

```
4
2
-
```

Output:

```
Enter the priority of denominations (1 for highest priority, 4 for
Enter priority for denomination 2000: Enter priority for denominat
Total balance available in the ATM machine is: 2061000
```

9. Write a program using choice to check
Case 1: Given string is palindrome or not
Case 2: Given number is palindrome or not

The screenshot shows the OneCompiler Python IDE interface. The editor on the left contains a Python script for checking palindromes. The script defines a `check_palindrome` function that returns `True` if a string is a palindrome and `False` otherwise. It prompts the user to choose between checking a string or a number, then takes input and prints the result. The right-hand pane shows the standard input (STDIN) with the value '1' and 'hello', and the program's output, which confirms that 'hello' is not a palindrome.

```
1- def check_palindrome(value):
2-     return value == value[::-1]
3-
4- print("Choose an option:")
5- print("1. Check if a string is a palindrome.")
6- print("2. Check if a number is a palindrome.")
7- choice = int(input("Enter your choice (1 or 2): "))
8-
9- if choice == 1:
10-     s = input("Enter a string: ").replace(" ", "").lower()
11-     print(f"{s} is a palindrome." if check_palindrome(s) else f"{s} is not a palindrome.")
12- elif choice == 2:
13-     num = input("Enter a number: ")
14-     print(f"{num} is a palindrome." if check_palindrome(num) else f"{num} is not a palindrome.")
15- else:
16-     print("Invalid choice.")
17-
```

STDIN

```
1
hello
```

put:

```
ose an option:
Check if a string is a palindrome.
Check if a number is a palindrome.
er your choice (1 or 2): Enter a string: 'hello' is not a palindrome.
```

10. Find the LCM and GCD of n numbers?

The screenshot shows the OneCompiler Python IDE interface. The editor on the left contains a Python script that calculates the LCM and GCD of a list of numbers. It uses the `math` module for the GCD function. The script prompts the user for the number of elements, takes input for each element, and then prints the GCD and LCM of the numbers. The right-hand pane shows the standard input (STDIN) with the values '2', '12', and '24', and the program's output, which displays the GCD as 12 and the LCM as 24.

```
1 import math
2
3 def find_gcd(nums):
4     gcd_result = nums[0]
5     for num in nums[1:]:
6         gcd_result = math.gcd(gcd_result, num)
7     return gcd_result
8
9 def find_lcm(nums):
10    lcm_result = nums[0]
11    for num in nums[1:]:
12        lcm_result = (lcm_result * num) // math.gcd(lcm_result, num)
13    return lcm_result
14
15 n = int(input("Enter the number of elements: "))
16
17 numbers = [int(input(f"Enter number {i+1}: ")) for i in range(n)]
18
19 gcd = find_gcd(numbers)
20 lcm = find_lcm(numbers)
21
22 print(f"GCD of the numbers is: {gcd}")
23 print(f"LCM of the numbers is: {lcm}")
24
```

2
12
24

Output:

```
Enter the number of elements: Enter number 1: Enter number 2: GCD of the numbers is: 12
LCM of the numbers is: 24
```