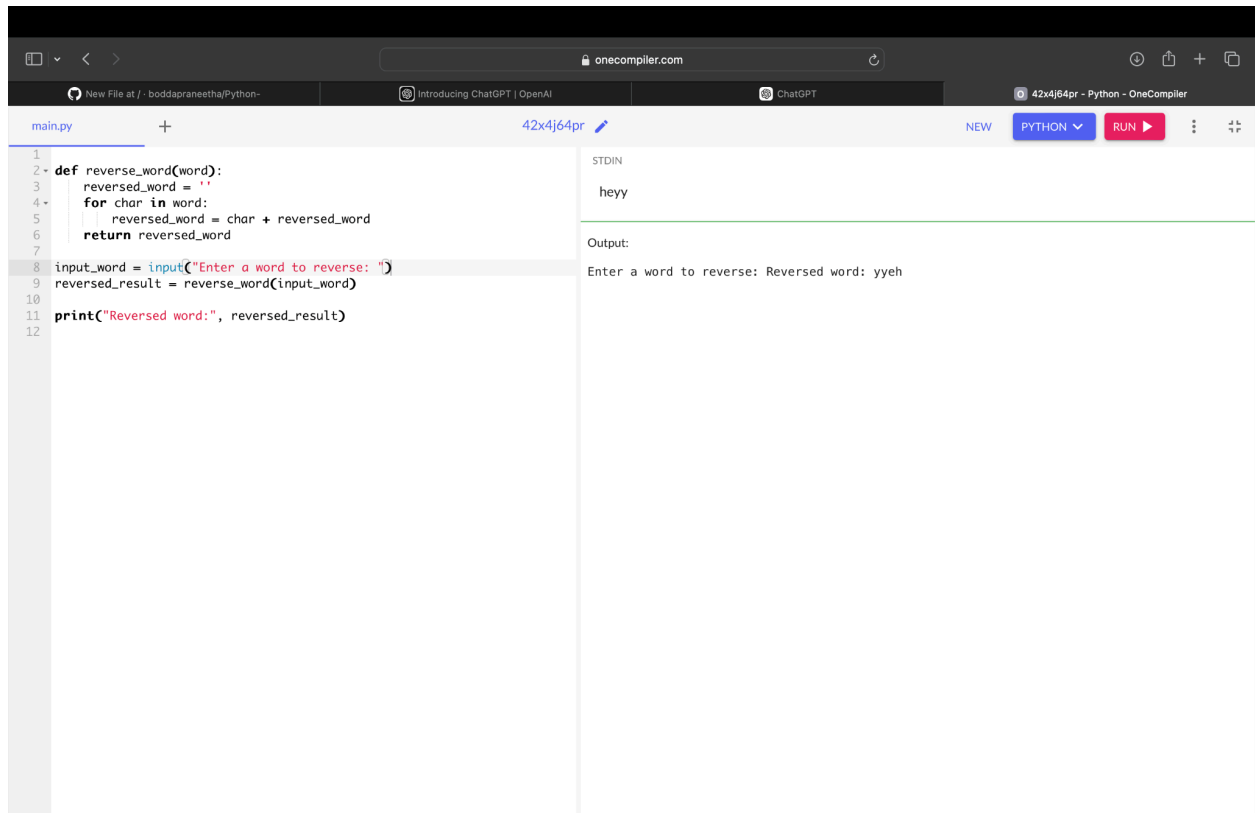


1. Write a python program to reverse a word using loop?

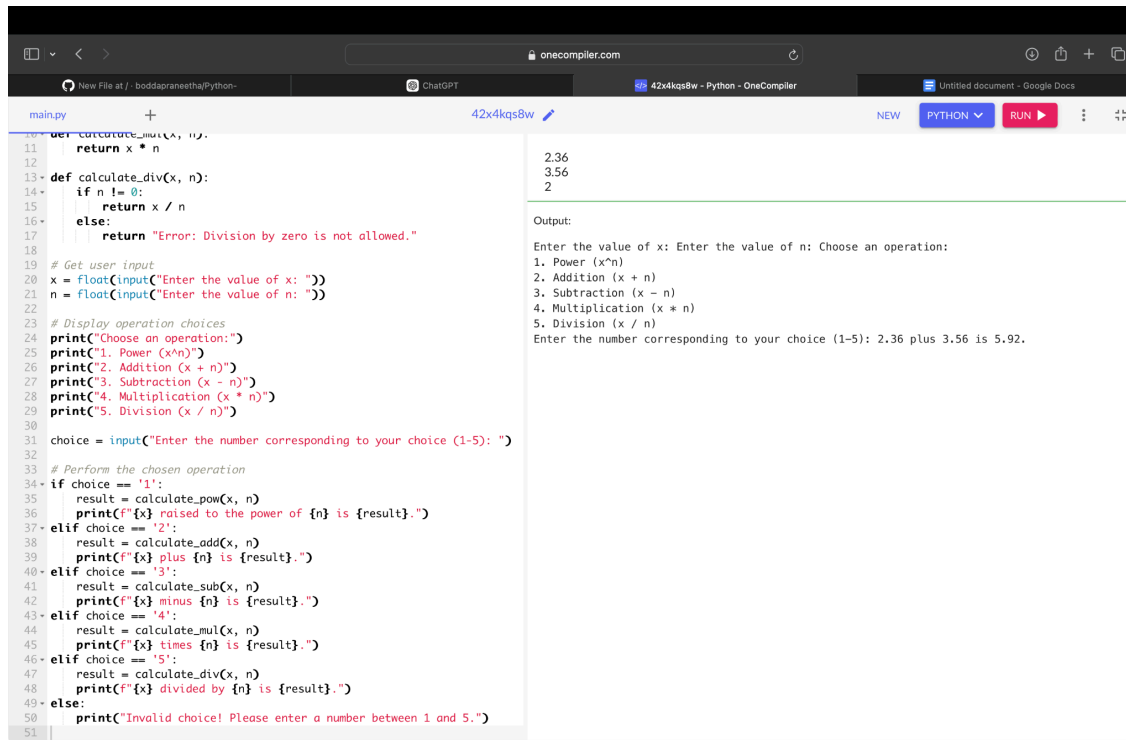


The screenshot shows the OneCompiler web interface. The browser address bar displays 'onecompiler.com'. The editor has a tab for 'main.py' and a file size of '42x4j64pr'. The Python code is as follows:

```
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
```

The 'STDIN' section shows the input 'hey'. The 'Output' section shows the result: '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.

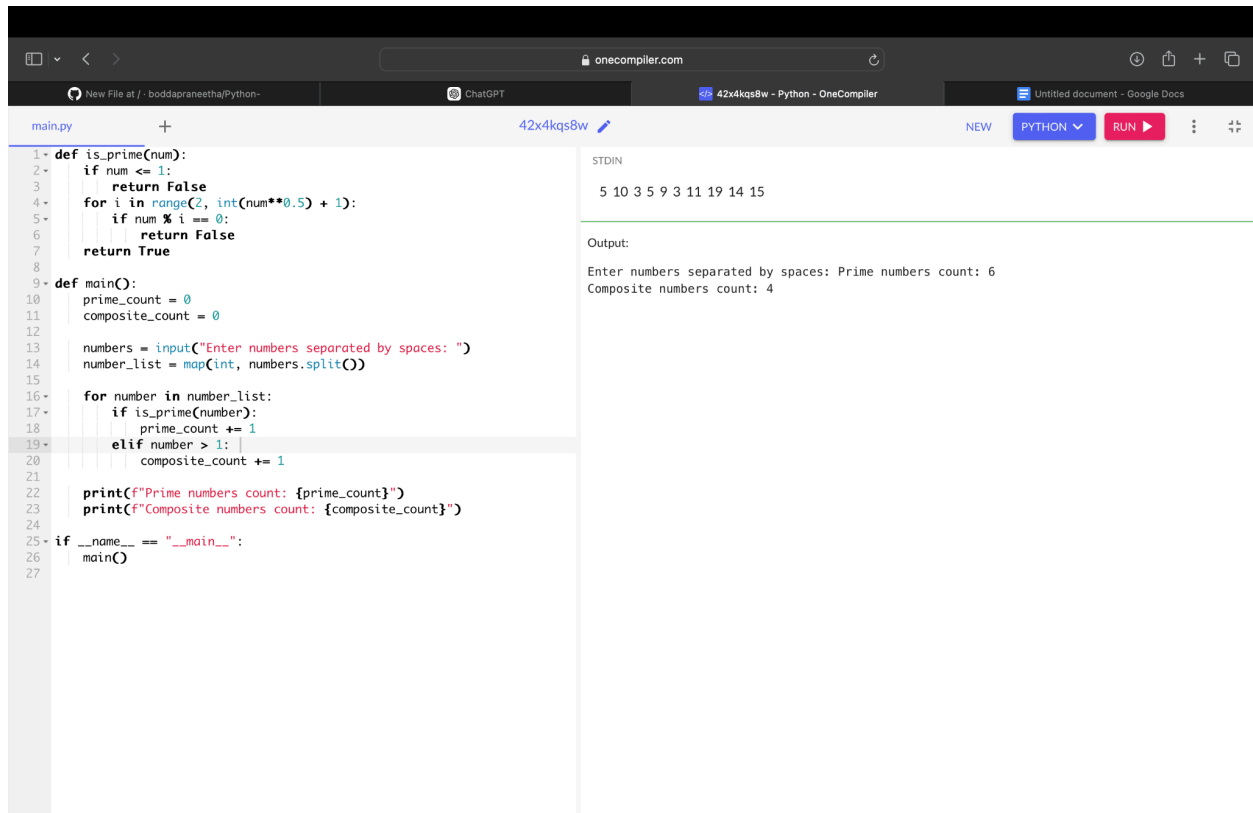


The screenshot shows a web browser window with the URL `onecompiler.com`. The browser tabs include "New File at / boddapaneetha/Python-", "ChatGPT", "42x4kqs8w - Python - OneCompiler", and "Untitled document - Google Docs". The main editor displays a Python file named `main.py` with the following code:

```
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
```

On the right side of the editor, there is a "Output" section. It shows the input values `2.36` and `3.56`, and the choice `2`. Below this, it displays the prompt "Enter the value of x: Enter the value of n: Choose an operation:" followed by a list of operations: "1. Power (x^n)", "2. Addition (x + n)", "3. Subtraction (x - n)", "4. Multiplication (x * n)", and "5. Division (x / n)". The final output line states: "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.



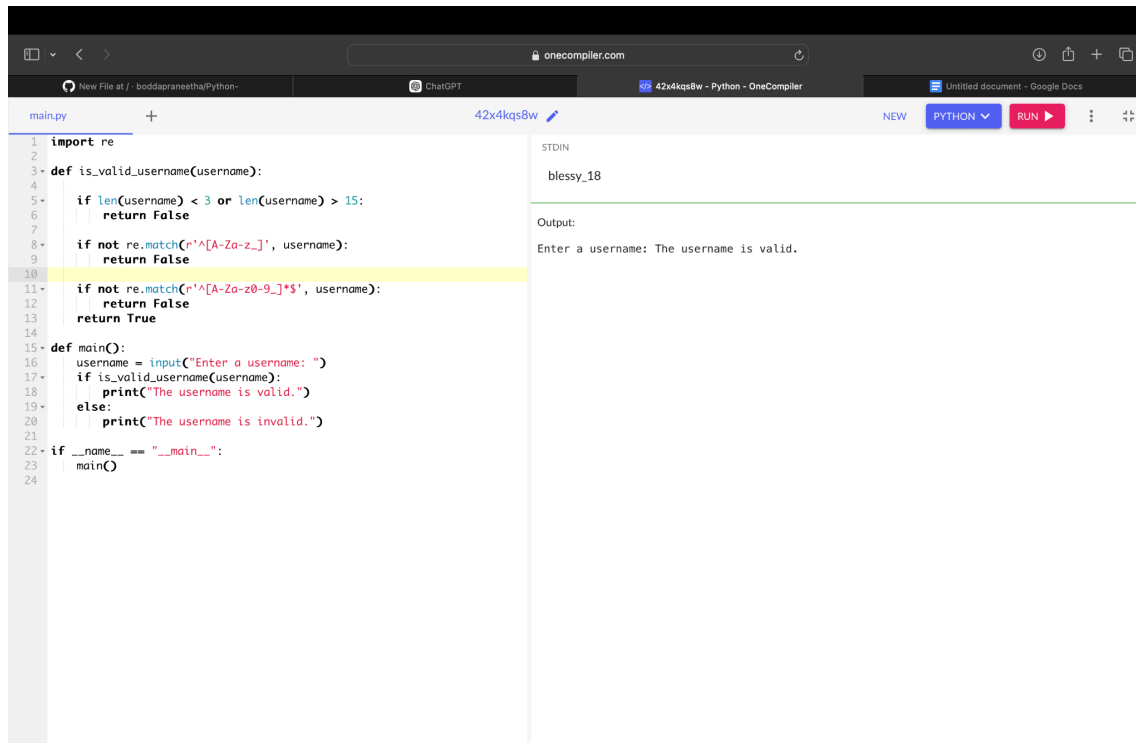
The screenshot shows the OneCompiler website interface. The left pane contains a Python script named `main.py` with the following code:

```
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-
```

The right pane shows the execution results. The `STDIN` input is `5 10 3 5 9 3 11 19 14 15`. The `Output` section displays the program's output:

```
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
5     if len(username) < 3 or len(username) > 15:
6         return False
7
8     if not re.match(r'^[A-Za-z_]', username):
9         return False
10
11     if not re.match(r'^[A-Za-z0-9_]*$', username):
12         return False
13     return True
14
15 def main():
16     username = input("Enter a username: ")
17     if is_valid_username(username):
18         print("The username is valid.")
19     else:
20         print("The username is invalid.")
21
22 if __name__ == "__main__":
23     main()
24
```

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.

The screenshot shows a Python program in the OneCompiler IDE. The program defines two functions: `find_mth_maximum` and `find_nth_minimum`. `find_mth_maximum` sorts the array in descending order and returns the m-th element. `find_nth_minimum` sorts the array in ascending order and returns the n-th element. The `main` function takes user input for the array, m, and n, calls these functions, and prints the results or an error message if the inputs are invalid.

```
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
6 def find_nth_minimum(arr, n):
7     unique_elements = sorted(set(arr))
8     return unique_elements[n - 1] if n <= len(unique_elements) else None
9
10
11 def main():
12     arr = list(map(int, input("Enter the numbers in the array separated by spaces: ").split()))
13     m = int(input("Enter the value of M for Mth maximum: "))
14     n = int(input("Enter the value of N for Nth minimum: "))
15
16     mth_max = find_mth_maximum(arr, m)
17     nth_min = find_nth_minimum(arr, n)
18
19     if mth_max is not None and nth_min is not None:
20         total = mth_max + nth_min
21         difference = mth_max - nth_min
22         print(f"{m}th Maximum Number: {mth_max}")
23         print(f"{n}th Minimum Number: {nth_min}")
24         print(f"Sum: {total}")
25         print(f"Difference: {difference}")
26     else:
27         print("Invalid M or N value. Please ensure they are within the range of the array length.")
28
29
30 if __name__ == "__main__":
31     main()
32
```

STDIN

```
123456789
2
4
```

Output:

```
Enter the numbers in the array separated by spaces: Enter the value
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 Python program in the OneCompiler IDE. The program takes a number as input and reverses it using a loop. It extracts the last digit using modulo 10, appends it to the reversed number (which is multiplied by 10), and then removes the last digit from the original number using integer division by 10.

```
1 number = int(input("Enter a number: "))
2 reversed_number = 0
3
4
5 while number != 0:
6
7     last_digit = number % 10
8     reversed_number = reversed_number * 10 + last_digit
9
10    number = number // 10
11
12 print("Reversed number:", reversed_number)
13
14
```

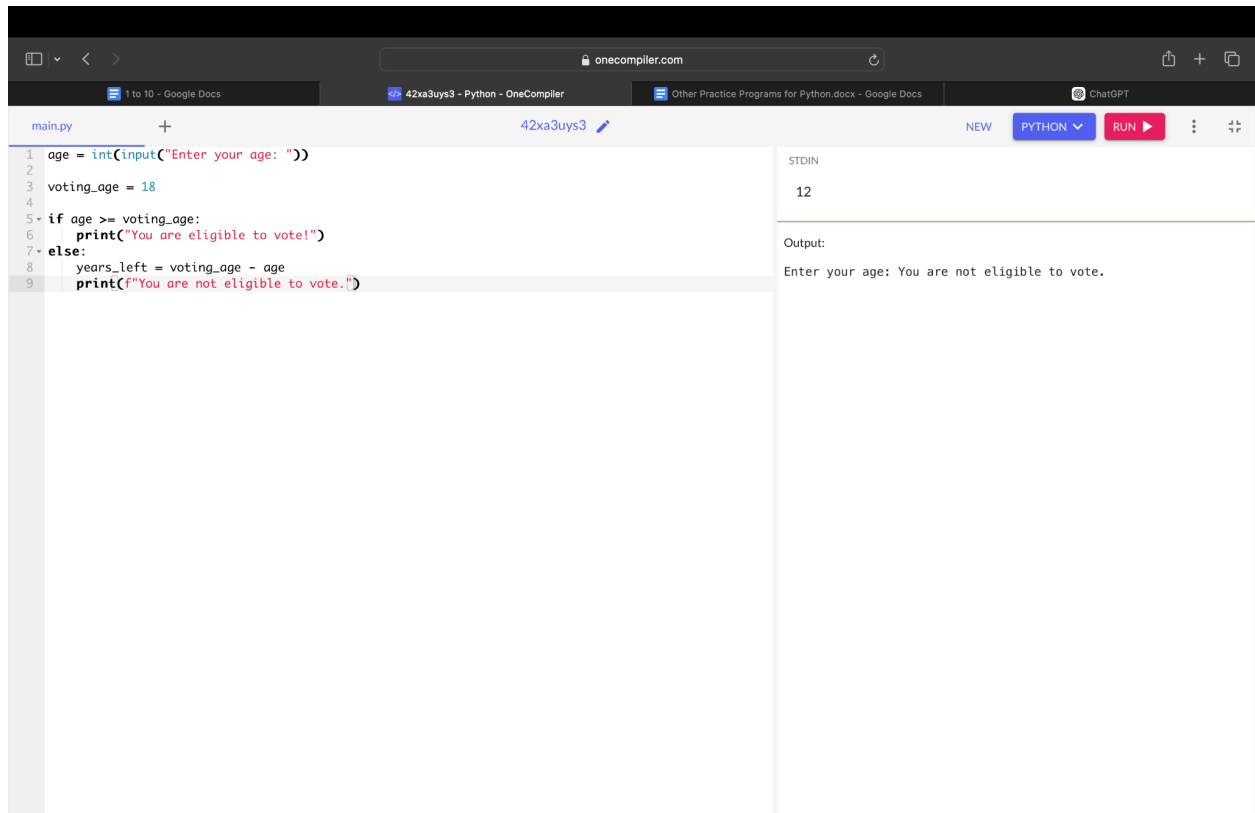
STDIN

```
1234
```

Output:

```
Enter a number: Reversed number: 4321
```

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.



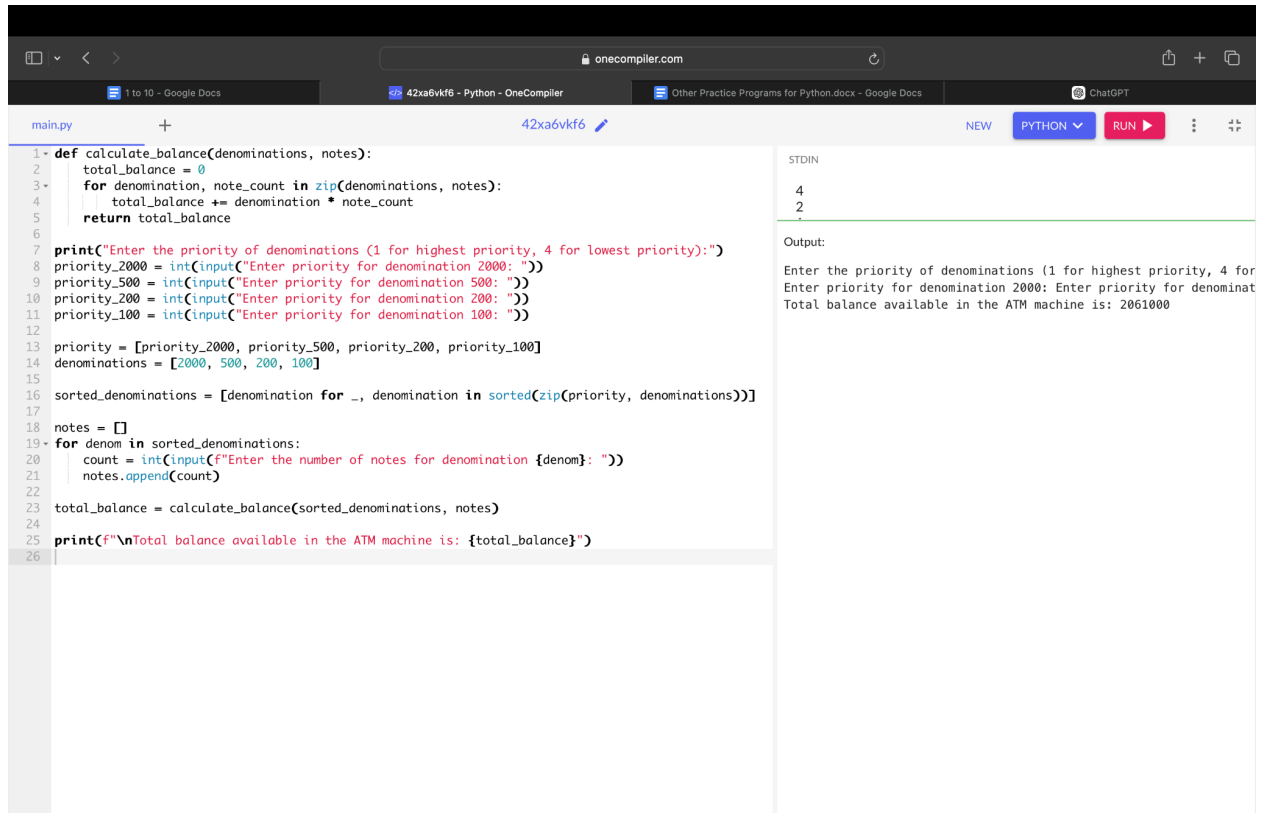
The screenshot shows a web browser window with the URL `onecompiler.com`. The browser has several tabs open, including "1 to 10 - Google Docs", "42xa3uys3 - Python - OneCompiler", and "Other Practice Programs for Python.docx - Google Docs". The active tab is "42xa3uys3 - Python - OneCompiler". The code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

On the right side of the editor, there are buttons for "NEW", "PYTHON", and "RUN". The "STDIN" input field contains the value "12". The "Output" section shows the result of the program execution:

```
Output:
Enter your age: You are not eligible to vote.
```

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-
```

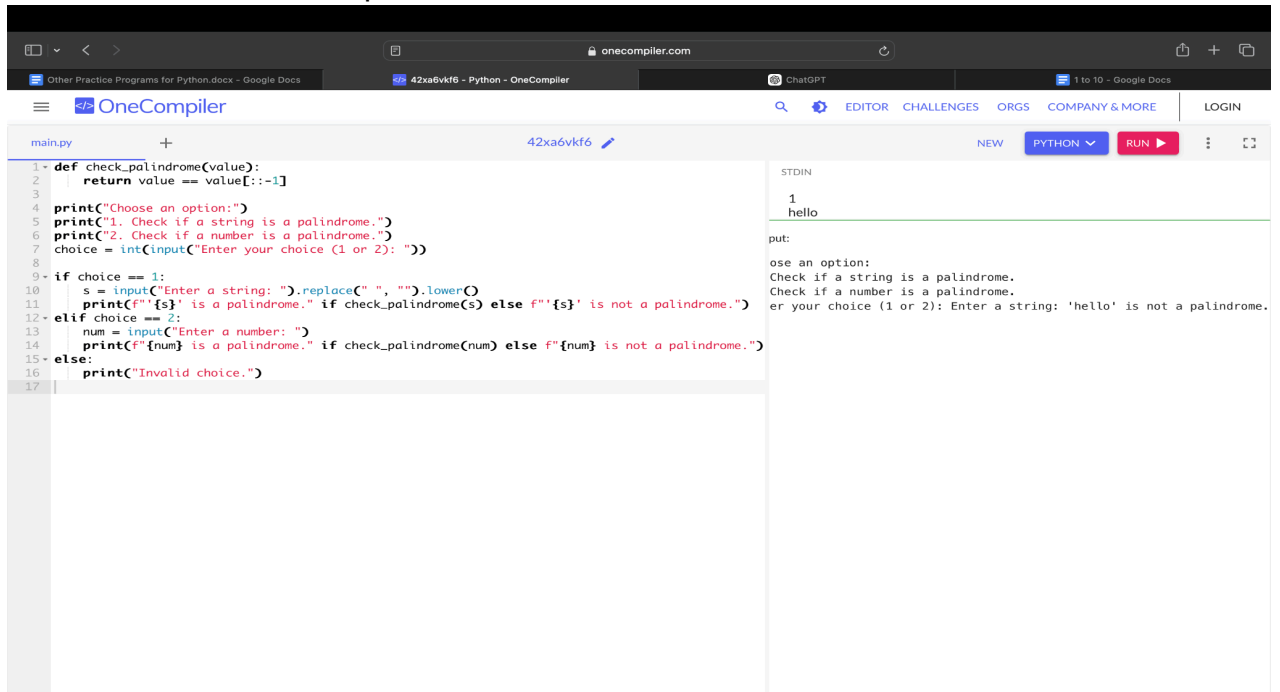
STDIN

```
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



```
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.
```