



main.py

```
1 #product has six months warranty. Write a program tocheck warranty of product by taking dat
2 def check_warranty(purchase_month, purchase_year):
3     warranty_expiry_month = (purchase_month + 5) % 12
4     warranty_expiry_year = purchase_year + (purchase_month + 5) // 12
5     if warranty_expiry_year < 2024 or (warranty_expiry_year == 2024 and warranty_expiry_mon
6         return True
7     elif warranty_expiry_year == 2024 and warranty_expiry_month == 3:
8         return True if 1 <= 25 else False
9     else:
10        return False
11 purchase_month = 9
12 purchase_year = 2023
13 is_warranty_valid = check_warranty(purchase_month, purchase_year)
14 if is_warranty_valid:
15     print("The warranty is still valid.")
16 else:
17     print("The warranty has expired.")
18
```



input

The warranty is still valid.

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1 #Write a function to take list and return multiplication of elements of list
2 def multiply_list_elements(lst):
3     result = 1
4     for num in lst:
5         result *= num
6     return result
7 my_list = [2, 3, 4, 5]
8 result = multiply_list_elements(my_list)
9 print(result)
10
11
```

```
120

...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1  #A list contain coupon code where last three characters of code represents cashback. Write a
2  def max_cashback_coupon(coupon_list):
3      max_cashback = 0
4      max_cashback_coupon = ""
5
6      for coupon in coupon_list:
7          cashback = int(coupon[-3:])
8          if cashback > max_cashback:
9              max_cashback = cashback
10             max_cashback_coupon = coupon
11
12     return max_cashback_coupon
13 coupon_list = ['chirstmas200', 'holiday120', 'Sunday100']
14 max_cashback = max_cashback_coupon(coupon_list)
15 print(f"The coupon code with maximum cashback is: {max_cashback}")
16
```



input

The coupon code with maximum cashback is: chirstmas200

...Program finished with exit code 0
Press ENTER to exit console.



Screenshot_2024-...



```
main.py
1 #product has six months warranty. Write a program to check warranty of product by taking data
2 def check_warranty(purchase_month, purchase_year):
3     warranty_expiry_month = (purchase_month + 5) % 12
4     warranty_expiry_year = purchase_year + (purchase_month + 5) // 12
5     if warranty_expiry_year < 2024 or (warranty_expiry_year == 2024 and warranty_expiry_mo
6         return True
7     elif warranty_expiry_year == 2024 and warranty_expiry_month == 3:
8         return True if 1 <= 25 else False
9     else:
10        return False
11 purchase_month = 9
12 purchase_year = 2023
13 is_warranty_valid = check_warranty(purchase_month, purchase_year)
14 if is_warranty_valid:
15     print("The warranty is still valid.")
16 else:
17     print("The warranty has expired.")
18
```

The warranty is still valid.

...Program finished with exit code 0
Press ENTER to exit console.



```
main.py
1  #A list contains both odd and even numbers. Write a function that takes list and return dict
2  def separate_odd_even(lst):
3      result_dict = {'odd': [], 'even': []}
4      for num in lst:
5          if num % 2 == 0:
6              result_dict['even'].append(num)
7          else:
8              result_dict['odd'].append(num)
9      return result_dict
10 my_list = [2, 9, 7, 8]
11 result = separate_odd_even(my_list)
12 print(result)
13
```

input

```
{'odd': [9, 7], 'even': [2, 8]}

...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1  #A dictionary contains three genres and three movies of each genre. Write a function to take
2  import random
3  def get_random_movie(genre, movies_dict):
4      if genre in movies_dict:
5          return random.choice(movies_dict[genre])
6      else:
7          return "Genre not found."
8  my_dict = {'Action': ['A', 'B', 'C'], 'Anime': ['D', 'E', 'F'], 'Horror': ['G', 'H', 'I']}
9  genre = 'Anime'
10 random_movie = get_random_movie(genre, my_dict)
11 print(f"A random movie from the {genre} genre is: {random_movie}")
12
```



input

A random movie from the Anime genre is: E

...Program finished with exit code 0
Press ENTER to exit console.



```
main.py
1  #Write to program to check palindrome or not using recursion.
2  def is_palindrome(string):
3      if len(string) <= 1:
4          return True
5      elif string[0] != string[-1]:
6          return False
7      else:
8          return is_palindrome(string[1:-1])
9
10 input_string = "radar"
11 result = is_palindrome(input_string)
12 if result:
13     print(f"The string '{input_string}' is a palindrome.")
14 else:
15     print(f"The string '{input_string}' is not a palindrome.")
16
```

input

The string 'radar' is a palindrome.

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1  #A dictionary contains three genres and three movies of each genre. Write a function to tak
2  import random
3  def get_random_movie(genre, movies_dict):
4      if genre in movies_dict:
5          return random.choice(movies_dict[genre])
6      else:
7          return "Genre not found."
8  my_dict = {'Action': ['A', 'B', 'C'],'Anime': ['D', 'E', 'F'],'Horror': ['G', 'H', 'I']}
9  genre = 'Anime'
10 random_movie = get_random_movie(genre, my_dict)
11 print(f"A random movie from the {genre} genre is: {random_movie}")
12
```

```
A random movie from the Anime genre is: E

...Program finished with exit code 0
Press ENTER to exit console.
```




main.py

```
1 #list contains ingredients of product. Avoid the product if it has both wheat and milk
2 def avoid_product(ingredients):
3     if 'wheat' in ingredients and 'milk' in ingredients:
4         return True
5     else:
6         return False
7 product1_ingredients = ['sugar', 'milk', 'chocolate']
8 product2_ingredients = ['sugar', 'wheat', 'chocolate']
9 product3_ingredients = ['sugar', 'wheat', 'milk']
10
11 print("Avoid product 1:", avoid_product(product1_ingredients))
12 print("Avoid product 2:", avoid_product(product2_ingredients))
13 print("Avoid product 3:", avoid_product(product3_ingredients))
14
```



input

Avoid product 1: False
Avoid product 2: False
Avoid product 3: True

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1  #Print the permutations of 'a', 'b', 'c' without repetition.Output['abc', 'acb', 'bac', 'bca', 'cab', 'cba']
2  from itertools import permutations
3
4  characters = ['a', 'b', 'c']
5  perms = list(permutations(characters))
6
7  result = [''.join(perm) for perm in perms]
8  print(result)
9
```

```
['abc', 'acb', 'bac', 'bca', 'cab', 'cba']

...Program finished with exit code 0
Press ENTER to exit console.
```