Run | Debug | Stop | Share | Save | { } Beautify | Language Python 3

main.py

```python
#A course contains internal assessment of 50 marks and external assessment of 50 each. A st
def is_passed(internal_marks, external_marks):
    if internal_marks >= 30 and external_marks > 20:
        return True
    else:
        return False

internal_marks = 35
external_marks = 25
result = is_passed(internal_marks, external_marks)

if result:
    print("The student has passed the course.")
else:
    print("The student has not passed the course.")

```

input

```
The student has passed the course.


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

Run | Debug | Stop | Share | Save | { } Beautify | Language Python 3

main.py

```python
#course contains internal assessment of 50 marks and external assessment of 50 each. A stud
def is_passed(internal_marks, external_marks):
    return internal_marks >= 30 and external_marks > 20
students_data = {
    "Alice": [35, 25],
    "Bob": [28, 30],
    "Charlie": [40, 15],
    "David": [30, 25]
}
passed_students = []
for student, marks in students_data.items():
    internal_marks, external_marks = marks[0], marks[1]
    if is_passed(internal_marks, external_marks):
        passed_students.append(student)
print("List of students who have passed the exam:")
for student in passed_students:
    print(student)

```

input

```
List of students who have passed the exam:
Alice
David


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

Run Debug Stop Share Save { } Beautify Language Python 3

main.py

```python
#In a debate there are two team - positive, negative:Ten applicants ['A', 'B', 'C', 'D', 'E'
import random
applicants = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']
random.shuffle(applicants)
positive_team = applicants[:5]
negative_team = applicants[5:]
print("Positive Team:", positive_team)
print("Negative Team:", negative_team)

```

input

```
Positive Team: ['D', 'J', 'G', 'B', 'I']
Negative Team: ['H', 'E', 'C', 'F', 'A']


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

Run Debug Stop Share Save { } Beautify Language Python 3

main.py

```python
#Write a function to return factorial of number using recursiondefined by lambda function
factorial = lambda n: 1 if n == 0 else n * factorial(n - 1)
num = 5
print("Factorial of", num, "is", factorial(num))

```

input

```
Factorial of 5 is 120


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

Run | Debug | Stop | Share | Save | {} Beautify | Language Python 3

main.py

```python
#Write a program to arrange the character of string in ascending order of their ASCII value
def arrange_chars_ascending(input_string):
    chars_list = list(input_string)
    sorted_chars = sorted(chars_list)
    sorted_string = ''.join(sorted_chars)

    return sorted_string
input_string = "hello"
result = arrange_chars_ascending(input_string)
print("String with characters arranged in ascending order of their ASCII values:", result)

```

input

```
String with characters arranged in ascending order of their ASCII values: ehllo


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

Run | Debug | Stop | Share | Save | { } Beautify | Language Python 3

main.py

```python
#Write a program to find difference(subtraction) betweenmaximum and minimum element of list
def find_difference(lst):
    if not lst:
        return "List is empty"
    max_element = max(lst)
    min_element = min(lst)
    difference = max_element - min_element
    return difference
my_list = [5, 10, 3, 8, 15]
result = find_difference(my_list)
print("Difference between maximum and minimum elements:", result)

```

input

```
Difference between maximum and minimum elements: 12


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

Run | Debug | Stop | Share | Save | {} Beautify | Language Python 3

main.py

```python
#Write a function to take list of two-digit numbers and round-
def round_to_nearest_ten(numbers):
    rounded_numbers = []
    for num in numbers:
        rounded_num = round(num, -1)
        rounded_numbers.append(rounded_num)

    return rounded_numbers
input_numbers = [23, 47, 58, 91, 36]
rounded_result = round_to_nearest_ten(input_numbers)
print("Original numbers:", input_numbers)
print("Rounded to nearest ten:", rounded_result)

```

input

```
Original numbers: [23, 47, 58, 91, 36]
Rounded to nearest ten: [20, 50, 60, 90, 40]


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