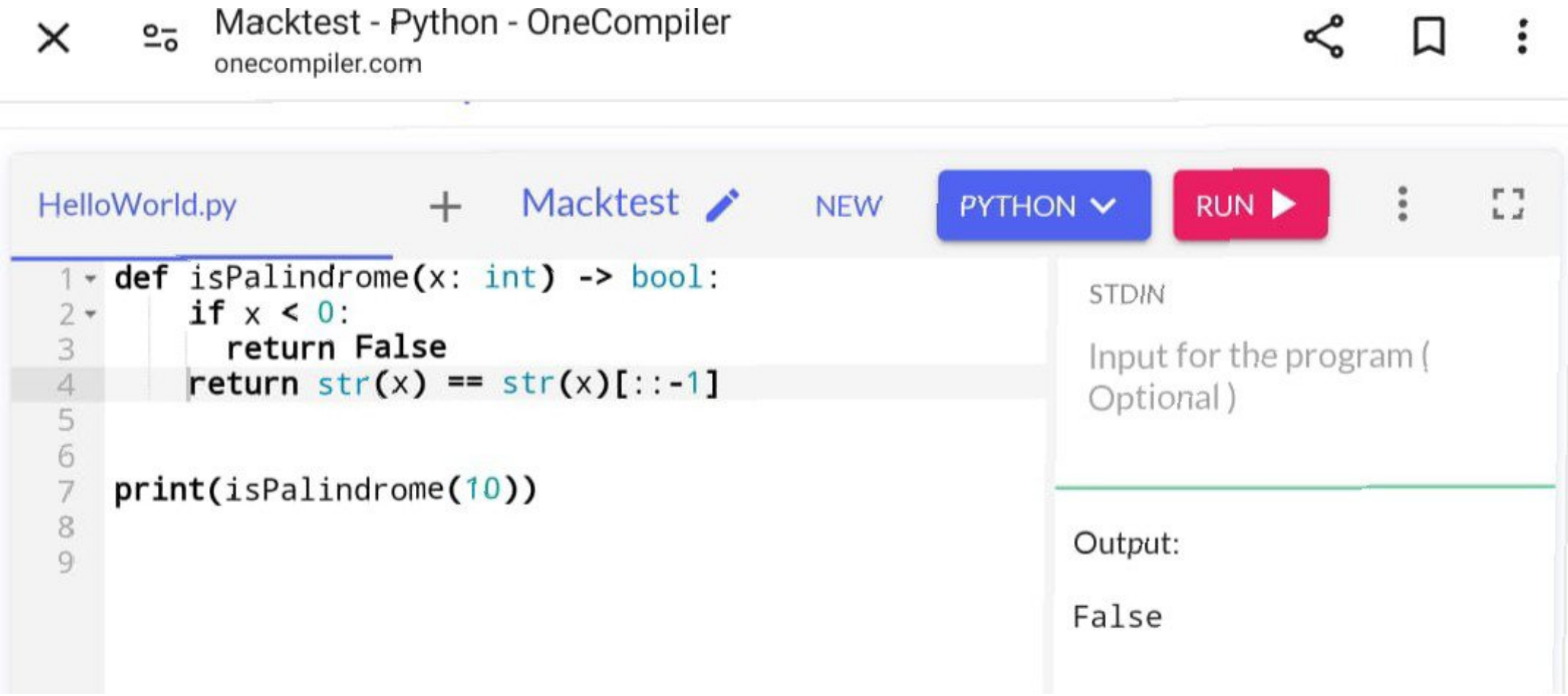
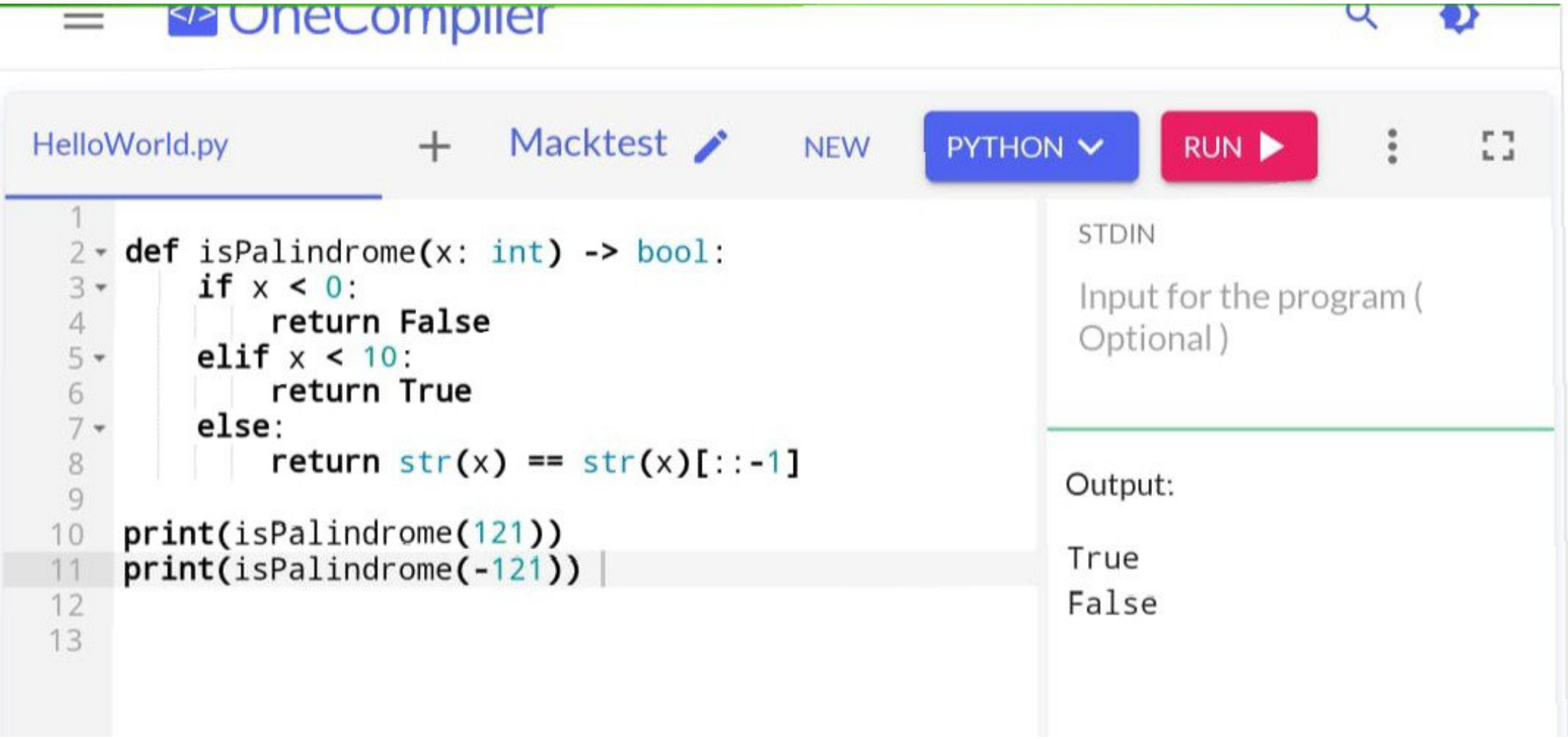

Macktest - Python - OneCompiler
onecompiler.com
HelloWorld.py
Macktest
NEW
PYTHON
RUN
def isPalindrome(x: int) -> bool:
if x < 0:
return False
return str(x) == str(x)[::-1]
print(isPalindrome(10))
STDIN
Input for the program (Optional)
Output:
False


23% 11:11
main.py
42cg45kjw
NEW
PYTHON
RUN
def two_sum(nums, target):
num_indices = {}
for i, num in enumerate(nums):
complement = target - num
if complement in num_indices:
return [num_indices[complement], i]
num_indices[num] = i
return []
nums = [2, 7, 11, 15]
target = 9
print(two_sum(nums, target))
STDIN
Input for the program (Optional)
Output:
[0, 1]


OneCompiler
HelloWorld.py
Macktest
NEW
PYTHON
RUN
def isPalindrome(x: int) -> bool:
if x < 0:
return False
elif x < 10:
return True
else:
return str(x) == str(x)[::-1]
print(isPalindrome(121))
print(isPalindrome(-121))
STDIN
Input for the program (Optional)
Output:
True
False

```python
def two_sum(nums, target):
    num_indices = {}
    for i, num in enumerate(nums):
        complement = target - num

        if complement in num_indices:
         return [num_indices[complement], i]
        num_indices[num] = i

    return []


nums = [3,3]
target = 6
print(two_sum(nums, target))
```

STDIN

Input for the program (Optional)

Output:

```
[0, 1]
```

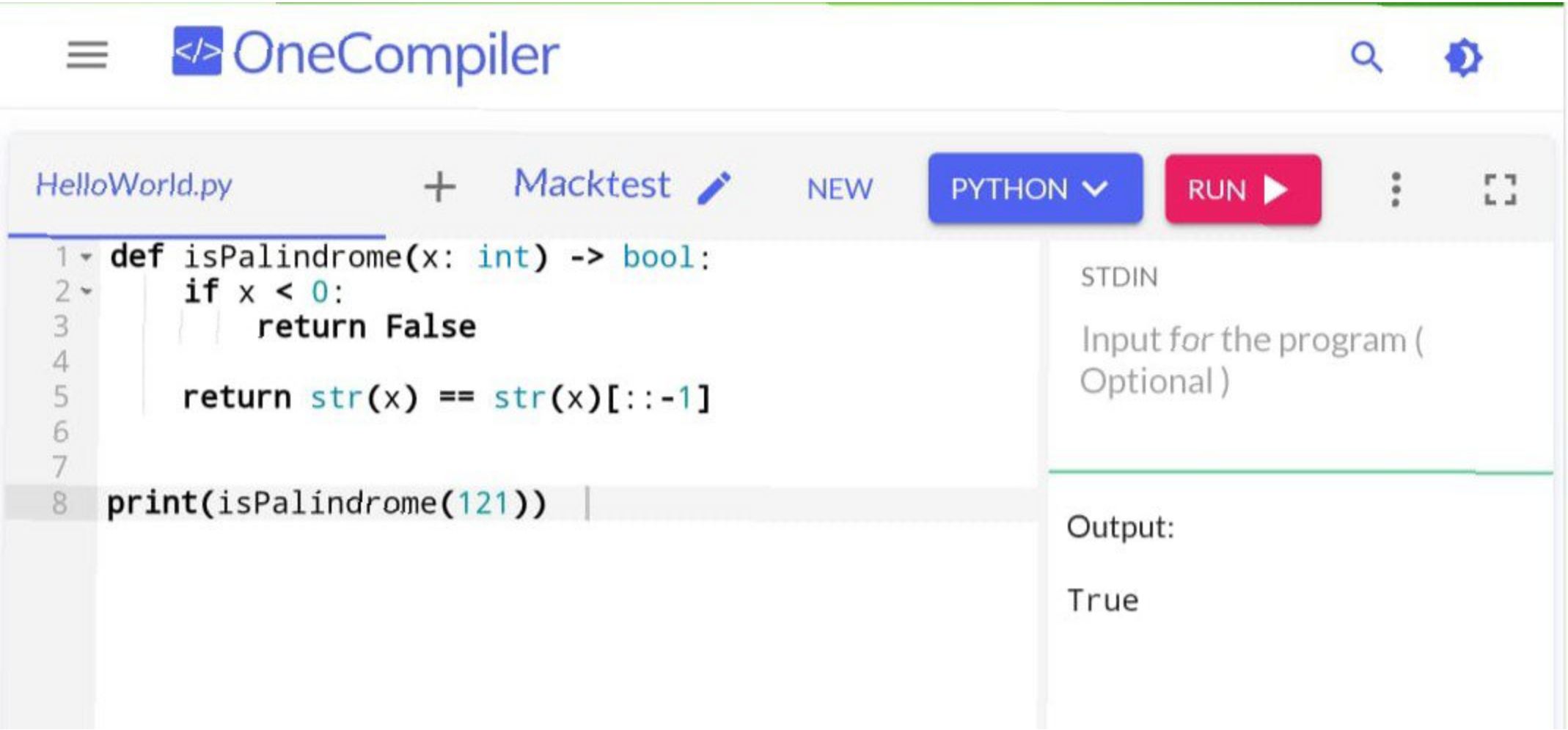
```python
def two_sum(nums, target):
    num_indices = {}
    for i, num in enumerate(nums):
        complement = target - num

        if complement in num_indices:
         return [num_indices[complement], i]
        num_indices[num] = i

    return []


nums = [3,2,4]
target = 6
print(two_sum(nums, target))
```

STDIN

Input for the program (Optional)

Output:

```
[1, 2]
```


OneCompiler
HelloWorld.py
Macktest
NEW
PYTHON
RUN
def isPalindrome(x: int) -> bool:
if x < 0:
return False
return str(x) == str(x)[::-1]
print(isPalindrome(121))
STDIN
Input for the program (Optional)
Output:
True