```python
x1 = input()
x2 = input()
y1 = input()
y2 = input()
y3 = input()
z = input()

# swap the values of `x1` and `x2`
x1, x2 = x2, x1

# do a circular swap of `y1`, `y2` and `y3`  like y1 = y2, y2 = y3, y3 = y1
y1, y2, y3 = y2, y3, y1

# create a new variable `a` with the value of `z`
a = z

# delete the variable `z`
del z

print(x1)
print(x2)
print(y1)
print(y2)
print(y3)
print(a)
```

```
# swap the values of `x1` and `x2`
x1, x2 = x2, x1

# do a circular swap of `y1`, `y2` and `y3`  like y1 = y2, y2 = y3, y3 = y1
y1, y2, y3 = y2, y3, y1

# create a new variable `a` with the value of `z`
a = z

# delete the variable `z`
del z
```

```python
main_dish = input()
time_of_day = int(input())
has_voucher = input().strip().lower() == "true"
is_card_payment = input().strip().lower() == "true"

if main_dish == "paneer tikka":
    cost = 250
elif main_dish == "butter chicken":
    cost = 240
elif main_dish == "masala dosa":
    cost = 200
else:  # if main dish is invalid print invalid dish and exit the code.
    print("Invalid main dish")
    exit()

if 12 <= time_of_day <= 15:
    total_cost = (1 - 0.15) * cost
else:
    total_cost = cost

if has_voucher:
    total_cost *= 0.9  # Apply voucher discount

if is_card_payment:  # service charge for card payments
    service_charge = 0.05 * total_cost
    total_cost += service_charge

print(f"{total_cost:.2f}")

```

```
45
46      elif operation == "vowel_check":
47          text = input().strip()
48          print(vowel_check(text))
49
50      elif operation == "divisibility_check":
51          number = int(input())
52          print(divisibility_check(number))
53
54      elif operation == "palindrominator":
55          text = input().strip()
56          print(palindrominator(text))
57
58      elif operation == "simple_interest":
59          principal_amount = int(input())
60          n_years = int(input())
61          print(simple_interest(principal_amount, n_years))
62
63      else:
64          print("Invalid Operation")
65
66  if __name__ == "__main__":
67      main()
```

```python
def perfect_square_check(number: int) -> str:
    root = int(math.sqrt(number))
    return "yes" if root * root == number else "no"

def vowel_check(text: str) -> str:
    vowels = "aeiouAEIOU"
    return "yes" if any(char in vowels for char in text) else "no"

def divisibility_check(number: int) -> str:
    if number % 2 == 0 and number % 3 == 0:
        return "divisible by 2 and 3"
    elif number % 2 == 0:
        return "divisible by 2"
    elif number % 3 == 0:
        return "divisible by 3"
    else:
        return "not divisible by 2 and 3"

def palindrominator(text: str) -> str:
    return text + text[-2::-1] if len(text) > 0 else text

def simple_interest(principal_amount: int, n_years: int) -> int:
    rate = 0.05 if n_years < 10 else 0.08
    interest = principal_amount * rate * n_years
```

Python3

```python

import math

# Multi-purpose application functions
def odd_num_check(number: int) -> str:
    return "yes" if number % 2 != 0 else "no"

def perfect_square_check(number: int) -> str:
    root = int(math.sqrt(number))
    return "yes" if root * root == number else "no"

def vowel_check(text: str) -> str:
    vowels = "aeiouAEIOU"
    return "yes" if any(char in vowels for char in text) else "no"
```

```python
# Validate the time
hour, period = time.split()
hour = int(hour)
is_time_valid = (1 <= hour <= 12) and (period in ["AM", "PM"])  # bool: True if time is val

# time_in_hrs:int should have the time in 24 hrs format . Try to do this in a single expres
time_in_hrs = (hour % 12) + (12 if period == "PM" else 0)

# time_of_day:str should have the time of the day as Morning, etc.. use "Invalid" if not wi
if not is_time_valid:
    time_of_day = "Invalid"
else:
    if period == "AM":
        if 6 <= hour < 12:
            time_of_day = "Morning"
        else:
            time_of_day = "Night"
    elif period == "PM":
        if hour == 12 or (1 <= hour < 6):
            time_of_day = "Afternoon"
        else:
            time_of_day = "Evening"

# write your code here
```

```
# part 1 - basic if

new_word = word # donot remove this line

# remove the "ing" suffix from `new_word` if it is there
if new_word.endswith("ing"):
    new_word = new_word[:-3]

# add the suffix "ing" to `new_word` if `continuous_tense` is True
if continuous_tense:
    new_word += "ing"

# write the whole if else block here

# part 2 - If else pattern

# age_group:str should be "Adult" or "Child" based on the age. assume age greater tha
if age >= 18:
    age_group = "Adult"
else:
    age_group = "Child"

# applicant_type:str should be age goup with the member status like "Adult Member" or
if is_member:
    applicant_type = f"{age_group} Member"
else:
    applicant_type = f"{age_group} Non-member"
```

```
42
43  # applicant_type:str should be age goup with the member status like "Adult Member" or "
44  if is_member:
45      applicant_type = f"{age_group} Member"
46  else:
47      applicant_type = f"{age_group} Non-member"
48
49  # write the whole if else block
50
51  # part 3 if ... elif .. else
52
53  # based on the value of `color_code` assign the `color` value in lower case and "black"
54
55  if color_code == "R":
56      color = "red"
57  elif color_code == "G":
58      color = "green"
59  elif color_code == "B":
60      color = "blue"
61  else:
62      color = "black"
63
64  # Validate the time
65  hour, period = time.split()
66  hour = int(hour)
67  is_time_valid = (1 <= hour <= 12) and (period in ["AM", "PM"])  # bool: True if time is
68
69  # time_in_hrs:int should have the time in 24 hrs format . Try to do this in a single ex
```

Python3

```python
# part 1 - If pattern
word = "glow" # str
continuous_tense = True # bool

# part 2
age = 5 # int
is_member = True # bool

# part 3

color_code = "R" # str: valid values are R-red, G-green and B-blue

time = "02 PM" # str, format:[2-digit hour][space][AM or PM]
# Morning (6 AM - 12 PM) (including the start and excluding the end)
# Afternoon (12 PM - 6 PM)
# Evening (6 PM - 12 AM)
# Night (12 AM - 6 AM)

# <eoi>

# part 1 - basic if

new_word = word # donot remove this line
```

Due Jun 19, 2024 at 11:59 PM GMT+5:30

Question | Test Cases | **Code** | Solution

Python3

```python
# A single quote ' and a double quote "
output1 = "A single quote ' and a double quote \""

# A forward slash / and a backward slash \
output2 = "A forward slash / and a backward slash \\"

# Three single quotes ''' and three double quotes """
output3 = "Three single quotes ''' and three double quotes \"\"\""

# Double forward slash // and Double backward slash \\
output4 = "Double forward slash // and Double backward slash \\\\"

```