3rdsept.control_flow

September 1, 2024

[4]:

```
#Q1.Write a Python program to check if a given number is positive or negative.
def check_number(n):
    if n<=0:
        return 'Negative'
    else:
        return 'Positive'
n=int(input('enter any number to check'))
to_check=check_number(n)
print(to_check)
```

```
enter any number to check -2
```

```
Negative
```

[13]:

```
#Q2.Create a program that determines if a person is eligible to vote based on␣
 ↪their age.
def vote_eligibilty(n):
    if n<=18:
        return 'You are not eligible to vote'
    else:
        return 'You are eligible to vote'
n=int(input('enter your age'))
to_check=vote_eligibilty(n)
print(to_check)
```

```
enter your age 20
```

```
You are eligible to vote
```

[14]:

```
#Q3. Develop a program to find the maximum of two numbers using if-else␣
 ↪statements.
def find_max(a,b):
    if a>b:
        return 'a is greater than b'
    else:
        return ' b is greater than a'
a,b=2,4
to_check=find_max(a,b)
print(to_check)
```

```
b is greater than a
```

[12]:

```
#Q4. Write a Python script to classify a given year as a leap year or not.
def check_leap(n):
    a=str(n)
    len_=len(a)
    if len_<4:
        return 'Enter correct year'
    elif n%4==0:
        return 'Given year is a leap year'
    else:
        return 'Given year is not a leap year'
n=int(input('Enter the year you want to check:'))
to_check=check_leap(n)
print(to_check)
```

```
Enter the year you want to check: 1996
```

```
Given year is a leap year
```

[15]:

```
#Q5.Create a program that checks whether a character is a vowel or a consonant.
def check_v_or_c(n):
    vowel='aeiouAeiou'
    if n != n.isalpha:
        return 'Enter any alphabet'
    else:
        if n in vowel:
            return 'Given character is a vowel'
        else:
            return 'Given character is a consonant'
n=input('Enter the character you want to check')
to_check=check_v_or_c(n)
print(to_check)
```

```
Enter the character you want to check 1
```

```
Enter any alphabet
```

[3]:

```
#Q6.Implement a program to determine whether a given number is even or odd.
def check_even_odd(n):
    if n%2==0 and n<0:
        return 'Given number is a negative even number'
    elif n%2==0 and n>0:
        return 'Given number is a positive even number'
    else:
        if n%2!=0 and n<0:
            return 'Given number is a negative odd number'
        elif n%2!=0 and n>0:
            return 'Given number is a positive odd number'
```

```
n=int(input('Enter the number you want to check'))
to_check=check_even_odd(n)
print(to_check)
```

```
Enter the number you want to check -25

Given number is a negative odd number
```

[78]:

```
#Q7.Write a Python function to calculate the absolute value of a number without␣
 ↪using the `abs()` function.
def absolute_value(n):
    if n< 0:
        return -(n)
    else:
        return n

n= int(input('Enter the number to check its absolute value'))
abs_value = absolute_value(n)
print("Absolute value of",n, "is", abs_value)
```

```
Enter the number to check its absolute value -5

Absolute value of -5 is 5
```

[3]:

```
#Q8.Develop a program that determines the largest of three given numbers using␣
 ↪if-else statements.
def check_greatest(a,b,c):
    if a>b and a>c:
        return 'Greatest is', a
    elif b>a and b>c:
        return 'Greatest is', b
    else:
        return 'Greatest is',c

a,b,c= -11,-2,-1
to_check=check_greatest(a,b,c)
print(to_check)
```

```
('Greatest is', -1)
```

[99]:

```
#Q9.Create a program that checks if a given string is a palindrome.
def is_palindrome(a):
    if a==a[::-1]:
        return 'String is palindrome'
    else:
        return 'String is not palindrome'
a=input('enter any string:')
```

```python
to_check=is_palindrome(a)
print(to_check)
```

```
enter any string: ayush
```

```
String is not palindrome
```

[6]:

```python
#Q10.Write a Python program to calculate the grade based on a student's score.
def check_grade(marks):
    if marks<60:
        return 'You are failed'
    elif 60<=marks<70:
        return ' You have got C grade'
    elif 70<=marks<80:
        return ' You have got B grade'
    else:
        if 80<=marks<=100:
            return ' You have got A grade'

marks=int(input('Enter the score'))
to_check=check_grade(marks)
print(to_check)
```

```
Enter the score 101
```

```
None
```

[2]:

```python
#Q11.Write a program to find the largest among three numbers using nested␣
 ↪if-else statements.
def check_greatest(a,b,c):
    if a>b and a>c:
        return 'Greatest is', a
    elif b>aand b>c:
        return 'Greatest is', c
    else:
        return 'Greatest is',b

a,b,c= -11,-2,-3
to_check=check_greatest(a,b,c)
print(to_check)
```

```
('Greatest is', -2)
```

[8]:

```python
#Q12.Implement a program to determine if a triangle is equilateral, isosceles,␣
 ↪or scalene.
def check_triangle(p,b,h):
    if p==b==h:
        return "Triangle with this measurement is equilateral"
    elif p==b!=h or b!=p==h:
        return "Triangle with this measurement is isosceles"
    else:
        if p!=b!=h:
            return "Triangle with this measurement is scalene"

p=input('Enter values of perpendicular:')
b=input('Enter values of base:')
h=input('Enter values of hypotenuse:')
to_check=check_triangle(p,b,h)
print(to_check)
```

```
Enter values of perpendicular: 22
Enter values of base: 30
Enter values of hypotenuse: 22
```

```
Triangle with this measurement is isosceles
```

[10]:

```
#Q13.Develop a program that checks if a year is a leap year and also if it is a␣
 ↪century year.
def check_leap(n):
    a=str(n)
    len_=len(a)
    if len_<4:
        return 'Enter correct year'
    elif n%4==0 and n%100==0:
        return 'Given year is a leap year and a century year'
    else:
        return 'Given year is not a leap year'
n=int(input('Enter the year you want to check'))
to_check=check_leap(n)
print(to_check)
```

```
Enter the year you want to check 2200
```

```
Given year is a leap year and a century year
```

[14]:

```
#Q14.Write a Python script to determine if a number is positive, negative, or␣
 ↪zero.
def check_number(n):
    if n<0:
        return 'Given number is a negative number'
    elif n==0:
        return n
    else:
        return 'Given number is a positive number'

n=int(input('Enter the number you want to check'))
```

```
to_check=check_number(n)
print(to_check)
```

```
Enter the number you want to check 2

Given number is a positive number
```

[1]:
```
#Q15.Create a program to check if a person is a teenager (between 13 and 19␣
 ↪years old).
def check_teen(n):
    if n<13:
        return 'The person is not a teen'
    elif 13<=n<=19:
            return 'The person is a teen'
    else:
        if n>19:
            return 'The person is not a teen'

n=float(input('Enter the age:'))
to_check=check_teen(n)
print(to_check)
```

```
Enter the age: 18.99

The person is a teen
```

[7]:
```
#Q16.Develop a program that determines the type of angle based on its measure␣
 ↪(acute, obtuse, or right).
def check_triangle(angle):
    if angle<90:
        return "It is an acute angle"
    elif angle==90:
        return "It is a right angle"
    else:
        if 180>angle>90:
            return "It is an obtuse angle"

angle=float(input('Enter values:'))

to_check=check_triangle(angle)
print(to_check)
```

```
Enter values: 45

It is an acute angle
```

[ ]:
```
#Q17. Write a Python program to calculate the roots of a quadratic equation.
```

[1]:

```python
#Q18.Implement a program to determine the day of the week based on a user-provided number (1 for Monday, 2
# for Tuesday, etc.).
day_number = int(input("Enter a number (1 for Monday, 2 for Tuesday, etc.): "))

days_of_week = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

if 1 <= day_number <= 7:
    day_name = days_of_week[day_number - 1]
    print(f"The day corresponding to the number {day_number} is {day_name}.")
else:
    print("Invalid input. Please enter a number between 1 and 7.")
```

```
Enter a number (1 for Monday, 2 for Tuesday, etc.):  3
```

```
The day corresponding to the number 3 is Wednesday.
```

[ ]:

```python
#Q19.Create a program that determines if a year is a leap year and also if it is evenly divisible by 400.
# 20. Develop a program that checks if a given number is prime or not using nested if-else statements.
# 21. Write a Python program to assign grades based on different ranges of scores using elif statements.
# 22. Implement a program to determine the type of a triangle based on its angles.
#REPEATED QUESTIONS
```

[7]:

```python
#Q23. Develop a program to categorize a given person's BMI into underweight, normal, overweight, or obese using
# elif statements.
given_BMI= float(input( 'enter your BMI'))
if given_BMI<=18.5:
    print('person is underweight')
elif 18.5<=given_BMI<=24.9:
    print('person has normal weight')
elif 25<=given_BMI<=29.9:
    print('person is overweight')
else:
    if given_BMI>=30:
        print('person is obese')

```

```
enter your BMI 22.2
```

```
person has normal weight
```

[ ]:

```
#Q24. Create a program that determines whether a given number is positive,␣
 ↪negative, or zero using elif
# statements.
# REPEATED QUESTIONS
```

[9]:

```
#Q25.Write a Python script to determine the type of a character (uppercase,␣
 ↪lowercase, or special) using elif
# statements.
a=input('enter any character')
if a.isupper():
    print('Given character is in uppercase')
elif a.islower():
    print('Given character is in lowercase')
else:
    print (' Given character is special character')
```

```
enter any character %
```

```
 Given character is special character
```

[16]:

```
# Q26. Implement a program to calculate the discounted price based on different␣
 ↪purchase amounts using elif
# statements.
def calculate_discount(a):
    if a in range(80,100):
        a = a - a*.05
        return a
    elif a in range(100,150):
        a=a-a*.10
        return a
    elif a in range(150,250):
        a=a-a*.15
        return a
```

[17]:

```
calculate_discount(80)
```

[17]:

```
76.0
```

[1]:

```
# 27. Develop a program to calculate the electricity bill based on different␣
 ↪consumption slabs using elif
# statements.

def calculate_electricity_bill(units):
    if units <= 100:
        bill = units * 1.5
    elif units <= 200:
        bill = (100 * 1.5) + (units - 100) * 2.5
    elif units <= 300:
        bill = (100 * 1.5) + (100 * 2.5) + (units - 200) * 4.0
    else:
        bill = (100 * 1.5) + (100 * 2.5) + (100 * 4.0) + (units - 300) * 6.0

    return bill

units_consumed = float(input("Enter the number of units consumed: "))

bill_amount = calculate_electricity_bill(units_consumed)

print(f"The electricity bill for {units_consumed} units is:  {bill_amount:.2f}")
```

```
Enter the number of units consumed:  40

The electricity bill for 40.0 units is:  60.00
```

```
[ ]:
```