**1.** yes

**2.** yes

**3.** no

مثبت سازی**4.**

**5.**

\*

در تقسیم و ضرب اگر هر دو وجود داشت اولویت با اونی ک چپ تره // /

%

در جمع و تفزیق اگر هر دو وجود داشت اولویت با اونی ک چپ تره + -

=

**6.**

(a) x

(b) x

(c) 2

(d) x+1

رشته با اعداد صحیح را نمیشه جمع کرد(e) error

(f) 3

**7.**

i1=2

i2=5

i3=-3

d1=2.0

d2=5.0

d3=-0.5

(a) print(i1+i2) 2+5=7

/ برای تقسیم اعشاری(b) print(i1/i2) 2/5=0.4

// برای تقسیم صحیح(c) 2//5=0

(d) 5/2=2.5

(e) 5//2=2

اول منفی سازی اعمال میشود و بعد ضرب (f) 2\*-3=-6

(g) 2.0+5.0=7.0

(h) 2.0 / 5.0 =0.4

(i) 5.0 / 2.0=2.5

(j) -0.5 \* 2.0 =-1.0

(k) 2.0 + 5= 7.0

(l) 2 / 5.0 = 0.4

(m) 5.0 / 2 =2.5

(n) 5 / 2.0 = 2.5

(o) 2/5\*2.0 = 0.8

(p) 2.0\*2/5 = 0.8

(q) 2.0/5.0\*2 = 0.8

(r) 2\*2.0/5.0 = 0.8

(s) 5/2\*2.0 = 5.0

(t) 2.0\*5/2 =5.0

(u) 5.0/2.0\*2 = 5.0

(v)2\*5.0/2.0 =5.0

چون / برای تقسیم اعشاری است**8**. 1.6666666666666667

**9.**

(a) 2+(5\*-3) = -13

(b) 2\*(5+(-3)) = 4

(c) 2/ 2 = 1.0

(d) 2// 2= 1

(e) 2 / 5 + -3 = -2.6

(f) 2 // 5+ -3 = -3

(g) 8.66666

(h) 8

(i) 4.0

(j) 4

(k) 2.0 + (5.0 \* -0.5) = -0.5

(l) 2.0 + 5.0 \* -0.5 = -0.5

(m) 2.0 / 5.0 + 0.5 = 0.9

(n) 2.0 / 5.5 = 0.36363

(o) 6.5 / 3 = 6.83333

(p) (2.0 +5.0 -0.5) /3 = 2.16666

(q) 2.0 + 5.0 +(-0.5/3) = 6.83333

(r) 3 \* 7.0 \* 2.5 = 52.5

**10.** #

**11.** The end of the line means the end of the comment, and to end the comment, you have to go to the next line

**12.** too few comments

**13.** The purpose of comments in programming is to provide additional information about the code that is not necessarily needed for the code to be executed, but which can help to explain what the code is doing, or provide additional context that may be useful for future developers who work on the code. Comments can also be used to temporarily disable certain parts of the code, allowing developers to test certain sections without having to completely delete the code.

**14.** Human readability is important in programming because it allows developers to quickly and easily understand the code they are working on. If code is not readable, it can be difficult to debug and maintain, and it can take a long time to figure out what the code is doing. Additionally, code that is easy to read is more likely to be shared and reused, which can save time and effort in the long run.

**15.**

**Name Error**: if we add letters with out define it we will see this error. for example : print(x)

This is because Python does not know what 'x' is and cannot assign a value to it

**Value Error:** this error Occurs when a value is passed to a function that the function is not designed to handle.

**Zero Division Error:** An example of a Zero Division Error arising in Python is when a number is divided by zero. For example, if you try to divide 10 by 0, Python will raise a Zero Division Error. This is because it is impossible to divide a number by zero, so Python is unable to execute the operation

.

**Indentation Error:** An example of an Indentation Error arising in Python is when the indentation of code is incorrect. For example, if you have an if statement with a code block that is not indented correctly, Python will raise an Indentation Error. This is because Python uses indentation to determine the structure of the code, so incorrect indentation will lead to an error

**Overflow Error:** An example of an Overflow Error arising in Python is when an arithmetic operation produces a result that is too large to be represented. For example, if you try to calculate the result of a very large number, Python may raise an Overflow Error. This is because Python has a limit on the size of numbers that it can represent, so if the result of an operation is too large, it will be unable to store the result and will raise an error

**Syntax Error:** Occurs when incorrect syntax is used in the code of an error-free Python program.

**Type Error**: An example of a Type Error arising in Python is when an operation or function is applied to an object of the wrong type. For example, if you try to add a string to an integer, Python will raise a Type Error. This is because Python cannot add a string to an integer, so it is unable to execute the operation with the given arguments**.**

**16.**

Lines 1, 2, 3, 5, and 8 are not wrong

There is a logical error in line 4, and to calculate the average, the sum of two numbers is first calculated, so parentheses are needed print((n1+n2)/2)

Line 6: there is a runtime exception. The denominator never equals zero (zeroDivisionError)

Line 7: syntax error d1=n1\*n2

**17.**

(a) x++

(b) x/=2

(c) x--

(d) x+=y

(e) x-=(y+7)

(f) x\*=2

(g) print ("number\_of\_closed\_cases + = 2\*ncc")

**18.**

3

1

print (x1) x1=2 x1+=1 x1=1+2

print(x2) x2=2 x2-=1 x2= 2-1

عملگر += یک را به مقدار x1 اضافه می کند، در حالی که -=

عملگر یک را از مقدار x2 کم می کند.

**19.**

(a) Because initially the radius is given a value of zere

(b) قبل از محاسبه محیط دایره. شعاع رو از کاربر دریافت کنیم

**20.**

برنامه ای بنویسید که طول و عرض مستطیل رو دریافت کرده و به ترتیب مساحت و محیط آن را نشان دهد

x=float(input("length"))

y=float(input("width"))

print (x \*y)

print ((x + y) \* 2)

**21.**

برنامه ای بنویسید که نام و نام خانوادگی و سن کاربر را دریافت کند

name1=input ("enter your first name : ")

name2=input ("enter your last name : ")

age =input ("how old are you? ")