Examen curs Python

1 point

```
s = "PythonCourse"
print(s[:-1:3])
```

Ce se va afisa la executia urmatorului cod:

- [ ] codul nu compileaza - nu se poate afisa o lista in python
- [ ] PtiCus
- [ ] Por
- [x] PhCr

1 point

```
print([x for x in range(1,10) if x in [1,3,5,7]])
```

Ce se va afisa la executia urmatorului cod:

- [x] [1, 3, 5, 7]
- [ ] [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- [ ] [2, 4, 6, 8]
- [ ] [3, 5, 7]
- [ ] [1, 2, 3, 4, 5, 6, 7, 8, 9]

1 point

```
d = {i%3 for i in [1,2,3,4,5,6,7,8,9,10] if i>5 }
print(d)
```

Ce se va afisa la executia urmatorului cod:

- [ ] {5, 8}
- [ ] {6, 9}
- [x] {0, 1, 2}
- [ ] {5, 6, 7}
- [ ] {6, 7, 8}
- [ ] {0, 3, 6, 9}

1 point

```
import re
print(re.sub("(\d\d)",lambda x: x.group(0)[0],"10+20=30"))
```

Ce se va afisa la executia urmatorului cod:

- [ ] 30
- [ ] 10+20=30
- [x] 1+2=3
- [ ] 20+10=30
- [ ] 0+0=0
- [ ] 10+30=20

1 point

```
x = [1,2,3,4,5,6,7,8,9,10]
y = list(map(lambda k: 10-k, x))
print(y)
```

Ce se va afisa la executia urmatorului cod:

- [ ] [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- [ ] [10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
- [ ] [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
- [x] [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
- [ ] [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- [ ] [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

1 point

```
l = list(sorted([1,25,73,17],key = lambda x: x/7))
print(l)
```

Ce se va afisa la executia urmatorului cod:

- [x] [1, 17, 25, 73]
- [ ] [73, 17, 25, 1]
- [ ] [1, 73, 25, 17]
- [ ] [73, 25, 17, 1]
- [ ] [17, 73, 25, 1]
- [ ] [25, 1, 17, 73]

1 point

```
x = float(11.0)
y = float(3.0)
print(x//y)
```

Ce se va afisa la executia urmatorului cod:

- [x] 3.0
- [ ] 4
- [ ] 3.66
- [ ] Eroare de compilare

1 point

```
import struct
s = ""
for i in struct.pack("@bhi",3,1,2):
        s+=hex(i)+" "
print(s)
```

Ce va afisa urmatorul cod ? (se ruleaza pe o masina Little Endian, x86, alignament standard)

- [ ] 0x3 0x1 0x0 0x0 0x2 0x0 0x0 0x0
- [ ] 0x3 0x0 0x0 0x1 0x2 0x0 0x0 0x0
- [ ] 0x3 0x1 0x2
- [ ] 0x3 0x1 0x2 0x0 0x0 0x0 0x0 0x0
- [x] 0x3 0x0 0x1 0x0 0x2 0x0 0x0 0x0

1 point

```
s = list(sorted({x for x in reversed("abracadabra")}))
print(s)
```

Ce se va afisa la executia urmatorului cod:

- [ ] ['r', 'd', 'c', 'b', 'a']
- [ ] ['a', 'r', 'b', 'a', 'd', 'a', 'c', 'a', 'r', 'b', 'a']
- [ ] ['a', 'a', 'a', 'a', 'a', 'b', 'b', 'c', 'd', 'r', 'r']
- [ ] ['a', 'b', 'r', 'a', 'c', 'a', 'd', 'a', 'b', 'r', 'a']
- [x] ['a', 'b', 'c', 'd', 'r']

1 point

```
s = "s=\"exec(s)\""
exec(s)
print(s)
```

Ce se va afisa la executia urmatorului cod:

- [x] exec(s)
- [ ] nu compileaza
- [ ] s=s
- [ ] s
- [ ] ruleaza intr-o bucla infinita

Back Submit