**SEMINARUL 2**

1. Se citesc două numere naturale a și b cu cel mult două cifre. Să se afișeze toate numerele naturale pozitive de cel mult două cifre care se divid cu 5 și nu se află în intervalul [a,b] (numerele se vor afișa pe aceeași linie, ordonate crescător/ descrescător).

a=int(input())  
b=int(input())  
**if** a>b:  
 a,b=b,a *#interschimbare*  
**for** i **in** range(5, a, 5):  
 print (i, end=**" "**) *#afisam numerele care se afla in stanga intervalului*bux=b+5-b%5  
**for** i **in** range(bux, 96, 5):  
 print (i, end=**" "**) *#afisam numerele care se afla in dreapta intervalului*

1. Se citește un șir de n >1 numere. Să se afișeze numărul de valori de tip deal (mai mari decât elementele vecine; primul și ultimul element din șir au doar un vecin).

n=int(input())  
x=int(input())  
y=int(input())  
nr=0  
**if** x>y:  
 nr=nr+1 *#cazul pentru primul element***for** i **in** range (2, n):  
 z=int(input())  
 **if** x<y & y>z :  
 nr=nr+1  
 x=y  
 y=z  
**if** x<y:  
 nr=nr+1 *#cazul pentru ultimul element*print (nr)

1. Se citește un șir de n numere. Să se determine lungimea maximă a unei subsecvențe crescătoare (strict) a acestui șir și o poziție de început a unei astfel de subsecvențe.

n=int(input())  
x=int(input())  
lg\_maxima=1  
lg=1  
poz\_final=0  
**for** i **in** range (1, n):  
 y=int(input())  
 **if** y>x :  
 lg=lg+1  
 **if** lg>lg\_maxima:  
 lg\_maxima=lg  
 poz\_final=i  
 **else**:  
 lg=1  
 x=y  
print (lg\_maxima)  
poz\_inceput=poz\_final-lg\_maxima+1  
print(poz\_inceput)

1. Se dau nouă numere naturale a și b. Să se afișeze cel mai mic număr Fibonacci din intervalul [a,b].

a=int(input())  
b=int(input())  
f1=1  
f2=1  
**while** f1<a:  
 f3=f1+f2  
 f1=f2  
 f2=f3  
**if** f1<=b:  
 print (f1)  
**else**:  
 print(**"nu exista"**)

1. Se dă un număr natural n. Sa se afișeze o descompunere a lui n ca sumă de termeni distincți din șirul lui Fibonacci care nu conține ca termeni două numere Fibonacci consecutive. Există mereu o astfel de descompunere?

n=int(input())  
a=b=1  
**while** b<=n:  
 a,b=b, a+b  
**if** b<=n:  
 print(b)  
 n=n-b  
**while** n>0:  
 **if** a<=n:  
 print (a, end=**" "**)  
 n=n-a  
 a,b=b-a,a

1. Se dă o sumă S și avem la dispoziție bancnote cu valorile: 1, 5, 10, 25. Să se determine o modalitate de a plăti suma S folosind un număr minim de bancnote.

Exemplu de ieșire pentru suma 123:

3 x bancnote de 1

2 x bancnote de 10

4 x bancnote de 25 (nu contează ordinea în care se afișează bancnotele din descompunere)

Declarare vector de bancnote: bancnote=[1,5,10,25]

bancnote=[1,5,10,25]  
bancnote.sort(reverse=**True**)  
frecventa=[0,0,0,0]  
s=int(input())  
**for** i **in** range(len(bancnote)):  
 **while** s-bancnote[i] >=0 :  
 s=s-bancnote[i]  
 frecventa[i]=frecventa[i]+1  
**for** i **in** range(len(frecventa)):  
 **if** frecventa[i]>0:  
 print(frecventa[i], **"x"**, bancnote[i])

1. Se citește un număr natural n.

a) Să se afișeze cel mai mare număr care se poate obține cu cifrele lui n

b) Să se afișează cel mai mic număr care se poate obține cu cifrele lui n (tema)

a)

n=input() *#citim numarul ca sir de caractere***for** i **in** range(9, -1, -1):  
 nr=n.count(str(i)) *#functia count pe sir de caractere* print(nr\*str(i), end=**''**)

b)

n=input() *#citim numarul ca sir de caractere*prima\_cifra\_nenula=0  
**for** i **in** range (1, 10):  
 nr=n.count(str(i))  
 **if** nr!=0:  
 print(str(i), end=**''**) *#afisam cea mai mica cifra nenula* prima\_cifra\_nenula=i  
 nr0=n.count(str(0)) *#numaram zerourile* print(nr0\*str(0), end=**''**) *#afisam zerourile* print((nr-1)\*str(i), end=**''**) *#afisam dupa zerouri, de nr-1 ori, cea mai mica cifra nenula* **break  
for** i **in** range(prima\_cifra\_nenula+1, 10):  
 nr=n.count(str(i))  
 print(nr\*str(i), end=**''**) *#afisam restul cifrelor de cate ori ne indica contorul*