

Zadaci za vježbu

Zadatak 1.

a.

```
File Edit Format Run Options Windows Help
# 05_01_a
def neparni(n):
    for i in range(n):
        if i % 2 != 0:
            yield i

n = int(input())
for t in neparni(n):
    print(t, end = ' ')
```

b.

```
File Edit Format Run Options Windows Help
# 05_01_b
def prost(n):
    for i in range(2, round(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def prosti(n):
    i = 2
    k = 0
    while k < n:
        if prost(i):
            k += 1
            yield i
        i += 1

n = int(input())
for t in prosti(n):
    print(t, end = ' ')
```

Zadatak 2.

```
File Edit Format Run Options Windows Help
# 05_02
def sljedeca(n):
    t = int(n, 2)
    t += 1
    t = bin(t)[2:]
    while len(t) < len(n):
        t = '0' + t
    return t

n = input()
print(sljedeca(n))
```

Zadatak 3.

```
File Edit Format Run Options Windows Help
# 05_03
from random import *
def nasumicna(n):
    a = randint(0, 2 ** n - 1)
    t = bin(a)[2:]
    while len(t) < n:
        t = '0' + t
    return t

n = int(input())
print(nasumicna(n))
```

5

Zadatak 4.

```
File Edit Format Run Options Windows Help
# 05_04
def sljedeca_permutacija(a):
    n = len(a) - 1
    while n >= 0 and a[n - 1] > a[n]:
        n -= 1
    n -= 1
    t = max(a[n:])
    k = n
    for i in range(n + 1, len(a)):
        if a[i] > a[n] and a[n] <= t:
            t = a[i]
            k = i
    a[n], a[k] = a[k], a[n]
    b = a[n + 1:]
    b.sort()
    a = a[:n + 1] + b
    return a

n = int(input())
a = [i + 1 for i in range(n)]
m = int(input())
for i in range(m - 1):
    a = sljedeca_permutacija(a)
for t in a:
    print(t, end = ' ')
```

5

Zadatak 5.

```
File Edit Format Run Options Windows Help
# 05_05
def sljedeca_permutacija(a):
    n = len(a) - 1
    while n >= 0 and a[n - 1] > a[n]:
        n -= 1
    n -= 1
    t = max(a[n:])
    k = n
    for i in range(n + 1, len(a)):
        if a[i] > a[n] and a[n] <= t:
            t = a[i]
            k = i
    a[n], a[k] = a[k], a[n]
    b = a[n + 1:]
    b.sort()
    a = a[:n + 1] + b
    return a

n = int(input())
a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])

t = 0
b = [i + 1 for i in range(n)]
while b != a:
    b = sljedeca_permutacija(b)
    t += 1
print(t)
```

5

Zadatak 6.

```
File Edit Format Run Options Windows Help
# 05_06
def sljedeca_permutacija(a):
    n = len(a) - 1
    while n >= 0 and a[n - 1] > a[n]:
        n -= 1
    n -= 1
    t = max(a[n:])
    k = n
    for i in range(n + 1, len(a)):
        if a[i] > a[n] and a[n] <= t:
            t = a[i]
            k = i
    a[n], a[k] = a[k], a[n]
    b = a[n + 1:]
    b.sort()
    a = a[:n + 1] + b
    return a

n = int(input())
a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])
b = [i + 1 for i in range(n)]
while b != a:
    t = b[:]
    b = sljedeca_permutacija(b)

for k in t:
    print(k, end = ' ')
```

Zadatak 7.

```
File Edit Format Run Options Windows Help
# 05_07
def fact(n):
    t = 1
    for i in range(1, n + 1):
        t *= i
    return t

def permutacije(a):
    yield a
    for br in range(fact(len(a)) - 1):
        n = len(a) - 1
        while n >= 0 and a[n - 1] > a[n]:
            n -= 1
        n -= 1
        t = max(a[n:])
        k = n
        for i in range(n + 1, len(a)):
            if a[i] > a[n] and a[n] <= t:
                t = a[i]
                k = i
        a[n], a[k] = a[k], a[n]
        b = a[n + 1:]
        b.sort()
        a = a[:n + 1] + b
    yield a

a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])
for t in permutacije(a):
    print(t)
```

Zadatak 8.

```
File Edit Format Run Options Windows Help
# 05_08
def podskup(k, n):
    t = bin(k)[2:]
    while len(t) < n:
        t = '0' + t
    st = '{'
    for i in range(len(t)):
        if t[i] == '1':
            st += str(i + 1) + ', '
    if len(st) > 1:
        st = st[:len(st) - 2]
    return st + '}'

n = int(input())
for i in range(2 ** n):
    print(podskup(i, n))
```

Zadatak 9.

```
File Edit Format Run Options Windows Help
# 05_09
def podskup(k, n):
    t = bin(k)[2:]
    while len(t) < n:
        t = '0' + t
    st = '{'
    for i in range(len(t)):
        if t[i] == '1':
            st += str(i + 1) + ', '
    if len(st) > 1:
        st = st[:len(st) - 2]
    return st + '}'

n = int(input())
m = int(input())
print(podskup(m - 1, n))
```

5

Zadatak 10.

```
File Edit Format Run Options Windows Help
# 05_10
def podskup(k, n):
    t = bin(k)[2:]
    while len(t) < n:
        t = '0' + t
    st = '{'
    for i in range(len(t)):
        if t[i] == '1':
            st += str(i + 1) + ', '
    if len(st) > 1:
        st = st[:len(st) - 2]
    return st + '}'

def rbrPodskupa(a, n):
    s = ''
    for i in range(1, n + 1):
        if i in a:
            s += '1'
        else:
            s += '0'
    return int(s, 2)

n = int(input())
a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])

print(podskup(rbrPodskupa(a, n) + 1, n))
```


5

Zadatak 11.

```
File Edit Format Run Options Windows Help
# 05_11
def podskup(k, n):
    t = bin(k)[2:]
    while len(t) < n:
        t = '0' + t
    st = '{'
    for i in range(len(t)):
        if t[i] == '1':
            st += str(i + 1) + ', '
    if len(st) > 1:
        st = st[:len(st) - 2]
    return st + '}'

def rbrPodskupa(a, n):
    s = ''
    for i in range(1, n + 1):
        if i in a:
            s += '1'
        else:
            s += '0'
    return int(s, 2)

n = int(input())
a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])

print(podskup(rbrPodskupa(a, n) - 1, n))
```

Zadatak 12.

```
File Edit Format Run Options Windows Help
# 05_12
def kombinacije(k, n):
    for j in range(2 ** n):
        t = bin(j)[2:]
        while len(t) < n:
            t = '0' + t
        st = '{}'
        r = 0
        for i in range(len(t)):
            if t[i] == '1':
                r += 1
                st += str(i + 1) + ', '
        if len(st) > 1:
            st = st[:len(st) - 2]
        if k == r:
            yield st + '}'

n = int(input())
k = int(input())
for t in kombinacije(k, n):
    print(t)
```

Zadatak 13.

```
File Edit Format Run Options Windows Help
# 05_13
from random import *

##generirat ćemo listu od k jedinica i n - k nula čije
##ćemo elemente ispremještati te ćemo na osnovu toga
##generirati kombinaciju
def kombinacija(k, n):
    t = [1] * k + [0] * (n - k)
    shuffle(t)
    st = '{}'
    for i in range(len(t)):
        if t[i] == 1:
            st += str(i + 1) + ', '
    if len(st) > 1:
        st = st[:len(st) - 2]
    return st + '}'

n = int(input())
k = int(input())
print(kombinacija(k, n))
```

5

Zadatak 14.

```
File Edit Format Run Options Windows Help
# 05_14
def particije(n, m = None):
    if m == None:
        m = n
    if (n == 0):
        yield []
    else:
        for i in range(1, min(n, m) + 1):
            for t in particije(n - i, i):
                yield [i] + t

n = int(input())
m = int(input())
for t in particije(n):
    if len(t) <= m:
        s = ''
        for k in t:
            s += str(k) + ' + '
        print(s[:len(s) - 3])
```

Zadatak 15.

```
File Edit Format Run Options Windows Help
# 05_15
def particije(n, m = None):
    if m == None:
        m = n
    if (n == 0):
        yield []
    else:
        for i in range(1, min(n, m) + 1):
            for t in particije(n - i, i):
                yield [i] + t

n = int(input())
m = int(input())
for t in particije(n):
    if max(t) == m:
        s = ''
        for k in t:
            s += str(k) + ' + '
        print(s[:len(s) - 3])
```