

## Приклад №12

### Створення багатомірного масиву та його індексація

```
example12.py > ...  
1  # Створення багатомірного масиву та його індексація  
2  
3  sub_list = [1, 2]  
4  sub_tuple = (5, 6)  
5  sub_dict = {8: 'a', 9: 'b', 10: 'c'}  
6  main_list = [sub_list, 3, 4, sub_tuple, 7, sub_dict]  
7  print(main_list)
```

```
>>> [[1, 2], 3, 4, (5, 6), 7, {8: 'a', 9: 'b', 10: 'c'}]
```

```
print(main_list[3])  
print(main_list[3][1])
```

```
>>> (5, 6)  
6
```

```
print(main_list[-1][8])
```

```
>>> a
```

```
main_dict = {  
    1: sub_list,  
    2: sub_tuple,  
    3: sub_dict  
}  
print(main_dict)
```

```
>>> {1: [1, 2], 2: (5, 6), 3: {8: 'a', 9: 'b', 10: 'c'}}
```

```
print(main_dict[3])  
print(main_dict[3][9])
```

```
>>> {8: 'a', 9: 'b', 10: 'c'}  
b
```

```
print(main_dict[2][0])
```

```
>>> 5
```

## Дії з багатомірними масивами

```
# Дії з багатомірними масивами

for i in main_list:
    if type(i) in (list, tuple, set):
        for k in i:
            print(k, end=" ")
    elif type(i) == dict:
        for k in i:
            print(i[k], end=" ")
    else:
        print(i, end=" ")
```

```
>>> 1 2 3 4 5 6 7 a b c
```

## Рекурсія

```
# Рекурсія

def recurs(arr):
    for i in arr:
        if type(i) in (list, tuple, set):
            recurs(i)
        elif type(i) == dict:
            for k in i:
                recurs(i[k])
        else:
            print(i, end=" ")

recurs((1, 2, main_list, sub_list, sub_dict))
```

```
>>> 1 2 1 2 3 4 5 6 7 a b c 1 2 a b c
```