Assignment 1

```
c = 0

def f(n):
    global c
    c += 1
    if n <= 1:
        return n
    return f(n-1) + f(n-2)

n = int(input())

for i in range(n):
    print(f(i), end=" ")

print("\nCalls:", c)
```

Assignment 2

```
def f1(n):
    if n==0:
        return 1
    return n*f1(n-1)

def f2(n):
    r=1
    for i in range(1,n+1):
        r*=i
    return r

n=int(input())
print("Recursion:",f1(n))
print("Iteration:",f2(n))
```

Assignment 3

```
j = [(1,4,20),(2,1,10),(3,1,40),(4,1,30)]

j.sort(key=lambda x:x[2], reverse=True)

s = [0]*4
p = 0

for a,b,c in j:
    for i in range(b-1,-1,-1):
        if s[i] == 0:
            s[i] = a
            p += c
            break

print("Job sequence:", s)
print("Total Profit:", p)
```

Assignment 4

```
w = [10,20,30]
p = [60,100,120]
c = 50

a = []
for i in range(3):
    a.append((p[i]/w[i], w[i], p[i]))

a.sort(reverse=True)

t = 0
for r,x,y in a:
    if c >= x:
        c -= x
        t += y
    else:
        t += r*c
        break

print("Maximum Profit:", t)
```

Assignment 5

```
w = [1,3,4,5]
p = [1,4,5,7]
W = 7
n = 4

d = [[0]*(W+1) for _ in range(n+1)]

for i in range(n):
    for j in range(W+1):
        if w[i] <= j:
            d[i+1][j] = max(p[i] + d[i][j-w[i]], d[i][j])
        else:
            d[i+1][j] = d[i][j]

print("Maximum Profit:", d[n][W])
```

Assignment 6

```
def C(n,r):
    if r==0 or r==n:
        return 1
    return C(n-1,r-1)+C(n-1,r)

n=int(input())
r=int(input())
print("Binomial Coefficient:",C(n,r))
```

Assignment 7

```
e = [(0,1,6),(0,2,7),(1,2,8),(1,3,5),(1,4,-4),
     (2,3,-3),(2,4,9),(3,1,-2),(4,0,2),(4,3,7)]

V = 5
d = [999]*V
d[0] = 0

for _ in range(V-1):
    for a,b,w in e:
        if d[a] != 999 and d[a] + w < d[b]:
            d[b] = d[a] + w

for i in range(V):
    print(0,"to",i,"=",d[i])
```

Assignment 8

```
import itertools

g = [
 [0,10,15,20],
 [10,0,35,25],
 [15,35,0,30],
 [20,25,30,0]
]

c = [1,2,3]
m = 99999

for p in itertools.permutations(c):
    t = g[0][p[0]]
    for i in range(2):
        t += g[p[i]][p[i+1]]
    t += g[p[2]][0]

    if t < m:
        m = t
        b = p

print("Best Path: 0 ->", b, "-> 0")
print("Minimum Cost:", m)
```

Assignment 9

```
g = [
 [0,2,0,1,0],
 [2,0,3,2,0],
 [0,3,0,0,1],
 [1,2,0,0,3],
 [0,0,1,3,0]
]
n = 5
d = [999]*n
v = [0]*n
d[0] = 0

for _ in range(n):
    u = -1
    for i in range(n):
        if not v[i] and (u == -1 or d[i] < d[u]):
            u = i

    v[u] = 1

    for j in range(n):
        if g[u][j] and not v[j]:
            if d[u] + g[u][j] < d[j]:
                d[j] = d[u] + g[u][j]
```

```
for i in range(n):
    print(0,"to",i,"=",d[i])
```

Assignment 10

```
e = [(1,2,1),(1,3,3),(2,3,2),(2,4,4),(3,4,5)]
p = {}

def f(x):
    if p[x] != x:
        p[x] = f(p[x])
    return p[x]

e.sort(key=lambda x: x[2])

for a,b,c in e:
    p[a] = a
    p[b] = b

cost = 0
print("Selected Edges:")
for a,b,c in e:
    if f(a) != f(b):
        p[f(a)] = f(b)
        cost += c
        print(a,"-",b,"=",c)

print("Total Cost:", cost)
```