

Zadatci za vježbu

Zadatak 1.

```
File Edit Format Run Options Windows Help
# 03_01
from Stack import *

s = input()
st = Stack()
uvjet = True
for i in range(len(s)):
    if s[i] == '(':
        st.push('(')
    elif s[i] == ')':
        if not st.isEmpty():
            st.pop()
        else:
            uvjet = False
if not st.isEmpty():
    uvjet = False
print(uvjet)
```

Zadatak 2.

```
File Edit Format Run Options Windows Help
# 03_02
from Stack import *

def igracke(izraz):
    s = Stack()
    ig = []
    for t in izraz:
        if t < 0:
            s.push(-t)
        else:
            if not t in ig:
                ig.append(t)
            if s.pop() != t:
                return None
    return ig

izraz = input().split()
for i in range(len(izraz)):
    izraz[i] = int(izraz[i])
t = igracke(izraz)
if t != None:
    t.sort()
    t.reverse()
    for k in t:
        print(k, end = ' ')
else:
    print(False)
```

Zadatak 3.

```
File Edit Format Run Options Windows Help
# 03_03
from Stack import *

def racunaj(br1, br2, op):
    if op == '+':
        return br1 + br2
    elif op == '-':
        return br1 - br2
    elif op == '*':
        return br1 * br2
    else:
        return br1 // br2

izraz = input()
s = izraz.split()
st = Stack()
for t in s:
    k = t
    if k in '+-*/':
        st.push(k)
    else:
        p = st.pop()
        while not st.isEmpty() and p not in '+-*/':
            k = str(racunaj(int(p), int(k), st.pop()))
            if (not st.isEmpty()):
                p = st.pop()
        st.push(p)
        st.push(k)

print(k)
```

Zadatak 4.

```
File Edit Format Run Options Windows Help
# 03_04
from Stack import *

izraz = input()
s = izraz.split()
s.reverse()
prefix = ''
st = Stack()
for t in s:
    if t in '+-*/()':
        if t == ')':
            st.push(t)
        elif t == '(':
            k = st.pop()
            while k != ')':
                prefix = k + ' ' + prefix
                k = st.pop()
        else:
            if not st.isEmpty():
                k = st.pop()
                if k == ')':
                    st.push(k)
                elif k in '+-' and t in '*/':
                    st.push(k)
                elif k == '+' and t == '-':
                    st.push(k)
                else:
                    prefix = k + ' ' + prefix
            st.push(t)
    else:
        prefix = t + ' ' + prefix
while not st.isEmpty():
    prefix = st.pop() + ' ' + prefix
print(prefix)
```

Zadatak 5.

```
# 03_05
from Stack import *

def kreiraj_postfix(izraz):
    s = izraz.split()
    postfix = ''
    st = Stack()
    for t in s:
        if t in '+-* /()':
            if t == '(':
                st.push(t)
            elif t == ')':
                k = st.pop()
                while k != '(':
                    postfix += k + ' '
                    k = st.pop()
            else:
                if not st.isEmpty():
                    k = st.pop()
                    if k == '(':
                        st.push(k)
                    elif k in '+-' and t in '*/':
                        st.push(k)
                    else:
                        postfix += k + ' '
                        st.push(t)
                else:
                    postfix += t + ' '
    while not st.isEmpty():
        postfix += st.pop() + ' '
    return postfix

def vrijednost_postfix(izraz):
    s = Stack()
    izraz = izraz.split()
    for t in izraz:
        if t in '+-* /':
            a = int(s.pop())
            b = int(s.pop())
            if t == '+':
                s.push(b + a)
            elif t == '-':
                s.push(b - a)
            elif t == '*':
                s.push(b * a)
            else:
                s.push(b // a)
        else:
            s.push(t)
    return s.pop()

izraz = input()
postfix = kreiraj_postfix(izraz)
print(vrijednost_postfix(postfix))
```

Zadatak 6.

```
File Edit Format Run Options Windows Help
# 03_06
from Queue import *
n = int(input())
r = Queue()
for i in range(1, n + 1):
    r.enqueue(i)
while not r.isEmpty():
    r.enqueue(r.dequeue())
    x = r.dequeue()
print(x)
```

Zadatak 7.

```
File Edit Format Run Options Windows Help
# 03_07
from Queue import *
n = int(input())
m = int(input())
r = Queue()
for i in range(1, n + 1):
    r.enqueue(i)

i = 0
while not r.isEmpty():
    i += 1
    x = r.dequeue()
    if i % m != 0:
        r.enqueue(x)
print(x)
```

Zadatak 8.

```
File Edit Format Run Options Windows Help
# 03_08
from Queue import *

n = int(input())
k = int(input())
m = int(input())
r1 = Queue()
for i in range(1, n + 1):
    r1.enqueue(i)
r2 = Queue()
r2.enqueue(1)
for i in range(n, 1, -1):
    r2.enqueue(i)
marica = []
ivica = []
ne_igraju = []
while len(ivica + marica + ne_igraju) < n:
    i = 0
    while i < k - 1:
        x = r1.dequeue()
        if not x in ivica + ne_igraju:
            i += 1
            r1.enqueue(x)
        x = r1.dequeue()
        i = 0
    while i < m - 1:
        y = r2.dequeue()
        if not y in marica + ne_igraju:
            i += 1
            r2.enqueue(y)
        y = r2.dequeue()
    if x != y:
        marica.append(x)
        ivica.append(y)
    else:
        ne_igraju.append(x)
for t in marica:
    print(t, end = ' ')
print()
for t in ivica:
    print(t, end = ' ')
```

Zadatak 9.

```
File Edit Format Run Options Windows Help
# 03_09
from Queue import *

n = int(input())
r = Queue()
p = set()
for i in range(1, 3 * n + 1):
    r.enqueue(i)
    p |= {i}
m = int(input())
s = [0] * 3
for i in range(3):
    s[i] = Queue()
for j in range(m):
    k = int(input())
    i = 0
    s[i].makeNull()
    while not r.isEmpty():
        x = r.dequeue()
        s[i % 3].enqueue(x)
        if i % 3 != k - 1:
            p -= {x}
        i += 1

    r.makeNull()
    for i in range(3):
        while not s[i].isEmpty():
            r.enqueue(s[i].dequeue())
if len(p) > 1:
    print('Nije jedinstvena')
else:
    print(list(p)[0])
```


Zadatak 10.

```
File Edit Format Run Options Windows Help
# 03_10
from Tree import *
r = None
rijec = input()
for c in rijec:
    if r == None:
        r = Tree(c)
    else:
        t = Tree(c)
        gotovo = False
        i = r
        while not gotovo:
            if i.V < t.V:
                if i.D == None:
                    i.D = t
                    gotovo = True
                else:
                    i = i.D
            else:
                if i.L == None:
                    i.L = t
                    gotovo = True
                else:
                    i = i.L
print(r.preorder())
```

Zadatak 11.

```
File Edit Format Run Options Windows Help
# 03_11
from Tree import *
r = None
preorder = input()
inorder = input()
for c in preorder:
    if r == None:
        r = Tree(c)
    else:
        t = Tree(c)
        i = r
        gotovo = False
        while not gotovo:
            x = inorder.index(i.V)
            y = inorder.index(t.V)
            if x < y:
                if i.D == None:
                    i.D = t
                    gotovo = True
                else:
                    i = i.D
            else:
                if i.L == None:
                    i.L = t
                    gotovo = True
                else:
                    i = i.L
print(r.postorder())
```

Zadatak 12.

```
File Edit Format Run Options Windows Help
# 03_12
from Tree import *
def bst(a):
    r = None
    for c in a:
        if r == None:
            r = Tree(str(c) + ' ')
        else:
            t = Tree(str(c) + ' ')
            gotovo = False
            i = r
            while not gotovo:
                if int(i.V) < int(t.V):
                    if i.D == None:
                        i.D = t
                        gotovo = True
                    else:
                        i = i.D
                else:
                    if i.L == None:
                        i.L = t
                        gotovo = True
                    else:
                        i = i.L
    return r

n = int(input())
a = []
for i in range(n):
    a.append(int(input()))
r = bst(a)
k = r.inorder()
k = k.split()
for t in k:
    print(t)
```

Zadatak 13.

```
File Edit Format Run Options Windows Help
# 03_13
from Tree import *
def bst(rijec):
    r = None
    for c in rijec:
        if r == None:
            r = Tree(c)
        else:
            t = Tree(c)
            gotovo = False
            i = r
            while not gotovo:
                if i.V < t.V:
                    if i.D == None:
                        i.D = t
                        gotovo = True
                    else:
                        i = i.D
                else:
                    if i.L == None:
                        i.L = t
                        gotovo = True
                    else:
                        i = i.L
            return r

def buc(r):
    if r.L != None or r.D != None:
        t = 1
        if r.L != None:
            t += buc(r.L)
        if r.D != None:
            t += buc(r.D)
    else:
        t = 0
    return t

rijec = input()
r = bst(rijec)
print(buc(r))
```

Zadatak 14.

```
File Edit Format Run Options Windows Help
# 03_14
from Tree import *
def bst(rijec):
    r = None
    for c in rijec:
        if r == None:
            r = Tree(c)
        else:
            t = Tree(c)
            gotovo = False
            i = r
            while not gotovo:
                if i.V < t.V:
                    if i.D == None:
                        i.D = t
                        gotovo = True
                    else:
                        i = i.D
                else:
                    if i.L == None:
                        i.L = t
                        gotovo = True
                    else:
                        i = i.L
            return r

def jed(r):
    if r.L != None or r.D != None:
        if r.L != None and r.D != None:
            t = 0
        else:
            t = 1
        if r.L != None:
            t += jed(r.L)
        if r.D != None:
            t += jed(r.D)
    else:
        t = 0
    return t

rijec = input()
r = bst(rijec)
print(jed(r))
```