**ALGORITMA DAN STRUKTUR DATA**

**MODUL 8**

**STACKS AND QUEUES**



**Disusun oleh:**

Muhammad Ferizal Fadhli

L200210119

D

**TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2022/2023**

Soal Soal Mahasiswa

1. Buatlah program untuk mengubah representasi suatu bilangan dari basis sepuluh ke basis dua.

Berikut kode program :

from latihan1 import Stack

def cetakHexa(angka):

    f = Stack()

    while angka != 0:

        sisa = angka % 16

        angka = angka // 16

        if sisa == 10:

            sisa = "A"

        elif sisa == 11:

            sisa = "B"

        elif sisa == 12:

            sisa = "C"

        elif sisa == 13:

            sisa = "D"

        elif sisa == 14:

            sisa = "E"

        elif sisa == 15:

            sisa = "F"

        f.push(sisa)

    st = ""

    for i in range(len(f)):

        st = st + str(f.pop())

    return st

print(cetakHexa(27))

Berikut Outputnya:  
 

1. Eksekusi program berikut, dan tunjukkan isi stacknya pada setiap Langkah.

Berikut kode program:

from latihan1 import Stack

nilai = Stack()

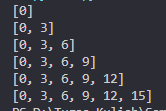
for i in range(16):

    if i % 3 == 0:

        nilai.push(i)

        print(nilai.**items**)

Berikut Outputnya:



1. Eksekusi program berikut, dan tunjukkan isi stacknya pada setiap Langkah.

Berikut kode program:

from latihan1 import Stack

nilai = Stack()

for i in range(16):

    if i % 3 == 0:

        nilai.push(i)

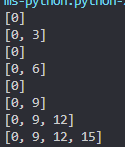
        print(nilai.**items**)

    elif i % 4 == 0:

        nilai.pop()

        print(nilai.**items**)

Berikut outputnya:



1. Tulis duaa metode berikut ke class Queue dan class PriorityQueue di atas.

* Metode untuk mengetehaui item yang paling depan tanpa menghapusnya.

def getFrontMost(self):

        assert not self.isEmpty(), "Antrian sedang kosong."

        return self.**qlist**[0].item

p1 = PriorityQueue()

p1.enqueue("Jeruk", 3)

print(p1.getFrontMost())

Berikut Output :



* Metode untuk mengetahui item yang paling belajang tanpa menghapusnya.

 def getRearMost(self):

        assert not self.isEmpty(), "Antrian sedang kosong."

        return self.**qlist**[-1].item

p1 = PriorityQueue()

p1.enqueue("Jeruk", 3)

p1.enqueue("Tomat", 4)

print(p1.getFrontMost())

print(p1.getRearMost())

Berikut output:  


1. Pada class priorityQueue metode dequeeue() belum diimplementasikan. Tulislah metodede dequeue() ini dengan memperhatikan syarat-syarat seperti yang telah dicantumkan.

Berikut kode yang telah dilengkapi :

    def dequeue(self):

        assert not self.isEmpty(), "Antrian sedang kosong."

        max = 0

        for i in range(len(self)):

            if self.**qlist**[i].priority < self.**qlist**[max].priority:

                max = i

        entry = self.**qlist**.pop(max)

        return entry.item