



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
Minggu ke / Materi	13 / Fungsi Rekursif

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

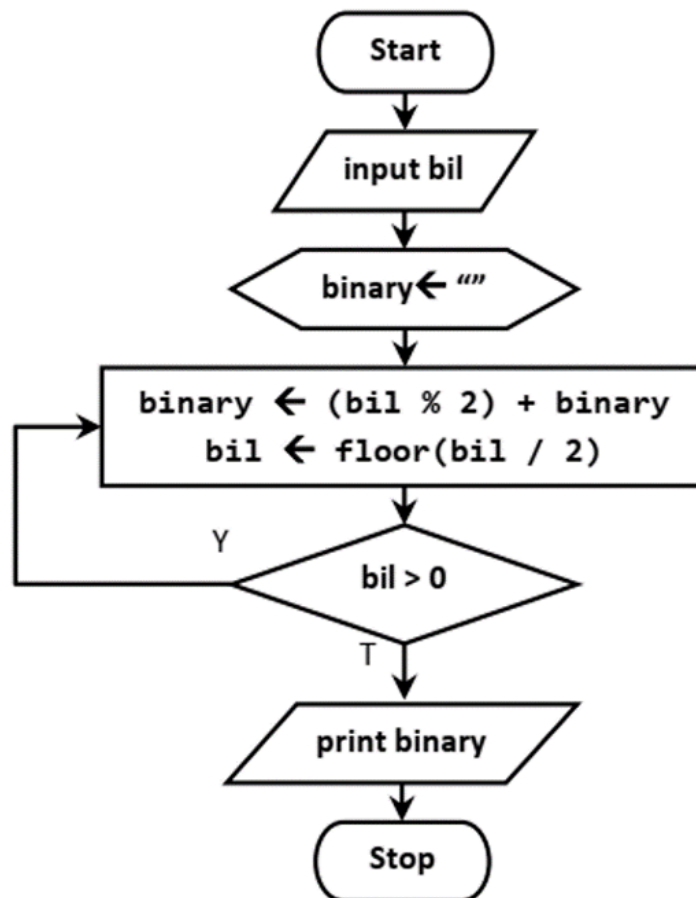
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

Link Github: <https://github.com/Tohrubo/tugaspralpro13.git>

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Rekursif

Rekursif adalah fungsi dalam python dimana sebuah fungsi akan memanggil fungsi itu sendiri untuk melanjutkan proses dengan data yang diubah sedikit. Sebuah fungsi rekursif harus mempunyai IF ELSE sebagai base case dan recursive case, jika tidak maka fungsi rekursif tidak dapat keluar dan hanya akan berulang terus menerus hingga crash atau error. Dibawah adalah contoh rekursi dalam bentuk flowchart:



Rekursi akan memanggil diri sendiri, dalam bentuk flowchart, bentuknya mirip seperti sebuah loop. Loop dan Recursion fungsionalitasnya mirip dan cara kejanya dapat digunakan untuk proses yang sama.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Welcome | Question1.py U x | Question2.py U | Question3.py U
Question1.py > chkprrec
1 def chkprrec(num, chk = 2):
2     if num == 2:
3         return True
4     elif num % chk == 0 or num == 1:
5         return False
6     elif int(num**0.5)+1 == chk:
7         return True
8     else:
9         return chkprrec(num, chk+1)
10
11 n = int(input("Number: "))
12 print(chkprrec(n))
```

B. Output Result

```
Number: 499
True
```

C. Explanation

Kode di atas menggunakan rekursi untuk menambah 1 ke variabel `chk` setiap rekursi. Lalu number akan dibagi `chk` di setiap rekursi, jika habis maka `False`, jika tidak maka akan masuk rekursi lagi. Jika `chk` mencapai akar dari number, maka sudah pasti bukan prima.

SOAL 2

A. Source Code

```
Welcome | Question1.py U | Question2.py U x | Qu
Question2.py > ...
1  def palchk(snt):
2      lst = list(snt)
3      if lst[0] == lst[-1]:
4          lst.pop(0)
5          lst.pop(-1)
6      else:
7          return "Not Palindrome"
8
9      if len(lst) < 2:
10         return "Palindrome"
11     else:
12         return palchk(lst)
13
14 sentence = input("Sentence: ")
15 sentence = sentence.lower()
16 print(palchk(sentence))
```

B. Output Result

```
Sentence: kasurRusak
Palindrome
```

C. Explanation

Kode diatas menggunakan list untuk memecah sebuah kalimat menjadi hurufnya saja. Kemudian akan mengecek apakah kata pertama dan terakhir sama, Jika sama maka akan dihapus dan masuk rekursi selanjutnya hingga habis. Jika tidak maka sudah pasti bukan palindrom.

SOAL 3

A. Source Code

```
Welcome | Question1.py U | Question2.py U | Question3.py U x | Question4.py U
Question3.py > oddco
1 def oddco(num, chk = 1, sc = 2):
2     if num == 0:
3         return 0
4     elif num == 1:
5         return chk
6     else:
7         return chk + oddco(num-1, chk + sc, sc*2)
8
9 n = int(input("To n = "))
10 print(oddco(n))
```

B. Output Result

```
To n = 5
57
```

C. Explanation

Kode di atas akan menggunakan 2 skalar, chk dan sc. chk untuk menyimpan hasil terbaru setelah dijumlah dengan sc. sc adalah skalar yang akan dikalikan 2 tiap rekursi. num digunakan untuk memastikan sampai ke n tersebut.

SOAL 4

A. Source Code

```
Welcome x Question1.py U Question2.py U C
Question4.py > ...
1 def numsum(num):
2     lst = list(num)
3     if len(lst) < 1:
4         return 0
5     else:
6         n = int(lst[0])
7         lst.pop(0)
8         return n + numsum(lst)
9
10 n = input("Numbers: ")
11 print(numsum(n))
```

B. Output Result

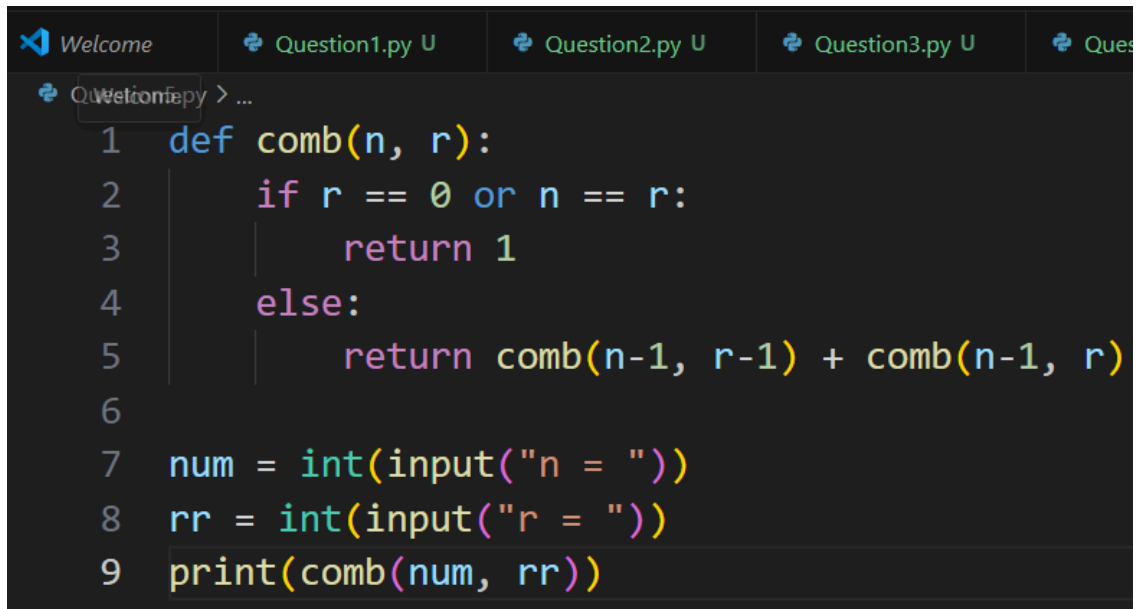
```
Numbers: 12345
15
```

C. Explanation

Kode di atas menggunakan list untuk memecah angka menjadi individual angka dalam sebuah list. Kemudian dimasukkan ke variabel n dan value di index 0 di pop supaya hilang, lalu n akan dijumlah fungsi rekursi hingga angka dalam list habis.

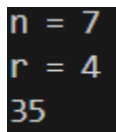
SOAL 5

A. Source Code



```
1 def comb(n, r):
2     if r == 0 or n == r:
3         return 1
4     else:
5         return comb(n-1, r-1) + comb(n-1, r)
6
7 num = int(input("n = "))
8 rr = int(input("r = "))
9 print(comb(num, rr))
```

B. Output Result



```
n = 7
r = 4
35
```

C. Explanation

Kode di atas menggunakan banyak rekursi untuk menghitung hasil kombinasi nCr . rekursi akan terjadi untuk setiap kemungkinan yang terjadi dari kombinasi, dan setiap 1 rekursi akan dihitung sebagai +1 hingga akhirnya mengeluarkan hasil total kombinasi.