

Modul Berpikir Komputasional - Python

Tim Materi Berpikir Komputasional 2024/2025

Catatan

1. Modul ini dirancang untuk dapat menjadi pegangan pemrograman Berpikir Komputasional, sehingga banyak hal yang dipotong karena keluar dari konteks Berpikir Komputasional.
2. Anda dapat membuka modul ini saat latihan praktikum.
3. Anda sangat disarankan untuk mencoba menjalankan semua program modul ini di komputer Anda, supaya Anda dapat mengetahui keluaran dari program yang ada.
4. Anda sangat disarankan untuk bereksperimen dari program-program yang ada di modul ini supaya Anda mendapat gambaran lebih jelas mengenai apa yang program Anda lakukan.
5. Anda sangat disarankan membaca tutorial dari tempat lain dan mengeksplor sendiri bahasa yang Anda gunakan.

Contents

1	Modul 1 - Input, Output, dan Percabangan	4
1.1	Pendahuluan	4
1.2	Input dan Output	4
1.3	Tipe Data	5
1.4	Operator	5
1.4.1	Operator Aritmatika	5
1.4.2	Operator Assignment	6
1.4.3	Operator Relasional	6
1.4.4	Operator Logika	6
1.5	Percabangan	6
1.6	Latihan Modul 1	8
1.6.1	Program Ganjil-Genap	8
1.6.2	Soal Praktikum 1 Tahun 2022	8
1.6.3	Soal Praktikum 2 Tahun 2022	8
1.6.4	Soal Praktikum 3 Tahun 2022	9
2	Modul 2 - Perulangan dan Array	10
2.1	Pengulangan	10
2.1.1	While Loop	10
2.1.2	For Loop	10
2.2	Perulangan bersarang	11
2.3	Array	11
2.3.1	Deklarasi Array	12
2.3.2	Array dan Variabel	12
2.3.3	String	12
2.4	Latihan Modul 2	14
2.4.1	Kelipatan 10 terkecil	14
2.4.2	Segitiga Angka	14
2.4.3	Reverse	14
2.4.4	Anagram	15
2.4.5	Palindrom	15
2.4.6	Soal Praktikum 1 Tahun 2021	16
2.4.7	Soal Praktikum 2 Tahun 2021	17
2.4.8	Soal Praktikum 3 Tahun 2021	17
2.4.9	Soal Praktikum 1 Tahun 2022	18
2.4.10	Soal Praktikum 2 Tahun 2022	18
3	Modul 3 - Fungsi dan Prosedur	19
3.1	Fungsi	19
3.2	Prosedur	19
3.3	Latihan Modul 3	20
3.3.1	Fungsi Kuadrat	20
3.3.2	Segitiga Pascal	21
3.3.3	Soal Praktikum 1 Tahun 2021	21

1 Modul 1 - Input, Output, dan Percabangan

1.1 Pendahuluan

Pada modul ini, kita menggunakan Python versi 3.9. Anda dapat mendownload Python di <https://www.python.org/downloads/release/python-390/>, pilihan "Windows x86-64 executable installer".

Beberapa karakteristik dari bahasa ini:

- Python bersifat *case-sensitive*, artinya perbedaan huruf besar dan huruf kecil menyebabkan perbedaan makna.
- Python sangat memerhatikan indentasi dan pergantian baris. Kesalahan indentasi dapat menyebabkan gagal *compile* hingga kesalahan program.
- Variabel di python bersifat implisit dan dinamis. Artinya, sebuah variabel tidak perlu dideskripsikan tipe datanya. Namun di modul ini kita tetap mempelajari tipe data yang ada.

1.2 Input dan Output

Dalam python, program untuk menulis "Hello, World!" ke layar adalah seperti berikut:

```
print("Hello, World!")           # (1)

print("Ini program", end="")     # (2)
print(" pertama saya")          # (3)

print("Dengan", "Koma")
# Ini akan menghasilkan luaran "Dengan Koma"
print("Dengan" + "Plus")
# Ini akan menghasilkan luaran "DenganPlus",
# Perhatikan '+' pada print melakukan konkatinasi pada string saja.

variabel = "sebuah nilai"
print(f"Dengan format {variabel}")
# Ini akan menghasilkan luaran "Dengan format sebuah nilai"
# {variabel} akan diubah dengan nilai dari variabel bersesuaian.

# Ini adalah sebuah komentar.

# Semua yg ada setelah tanda pagar (#)
# akan diabaikan oleh interpreter.

"""
Selain itu, kita juga bisa membuat komentar
multi baris dengan tiga petik (""")
"""
```

Bagian yang ditandai nomor 1 bertugas menuliskan "Hello, World!" ke layar. Sintaks `print` akan otomatis memindahkan baris setelah selesai menulis ke layar, kecuali disertai parameter `end=""`. Berarti, output program di atas adalah "Hello, World!" diikuti dengan "Ini program pertama saya" di baris selanjutnya.

Untuk melakukan input, kita membutuhkan penampung untuk menyimpan data yang diinputkan. Sebagai contoh, di bawah ini adalah program yang menerima input dan menuliskan ulang yang dimasukkan.

```
S = input("Masukkan kalimat: ") # 1
print("Anda memasukkan kalimat " + S) # 2

N = int(input("Masukkan sebuah angka: ")) # 3
print("Jika angka Anda ditambah 5, hasilnya " + str(N + 5)) # 4
```

Pada bagian nomor (1) dan (3), program membaca input dari user dan dimasukkan ke variabel S dan N. Lalu, bila ingin program menerima sebuah bilangan (sehingga bisa dijumlah / dikali / lainnya), kita bungkus `input()` dengan `int()`. Selain `int`, `string` (kumpulan karakter), Python dapat menerima tipe data `float` (bilangan real), dan masih banyak lagi.

1.3 Tipe Data

Ada beberapa macam tipe data, namun dalam Berpikir Komputasional kita hanya akan banyak menggunakan tipe data berikut:

<code>bool</code>	Boolean	True atau False
<code>int</code>	Bilangan bulat	seluruh bilangan bulat
<code>float</code>	Bilangan real	seluruh bilangan real
<code>string</code>	Teks	kumpulan karakter

Contoh penggunaan:

```
B = True # Boolean
I = 123456789012 # Bilangan bulat
R = 2.331973 # Bilangan real
S = "abc" # Teks
S = 'def'
```

Perhatikan pada variabel S, Anda bebas menggunakan petik satu (') atau petik dua (")

1.4 Operator

1.4.1 Operator Aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	2 + 3 bernilai 5
-	Pengurangan	1 - 8 bernilai -7
*	Perkalian	5 * 6 bernilai 30
/	Pembagian	13 / 5 bernilai 2.6
//	Pembagian (dibulatkan ke bawah)	13 // 5 bernilai 2
%	Sisa Bagi / Modulo	13 % 5 bernilai 3

1.4.2 Operator Assignment

Operator	Deskripsi	Contoh
=	Assignment	N = 5
+=	Penjumlahan	N += 5, N akan ditambah 5.
-=	Pengurangan	N -= 5, N akan dikurang 5.
*=	Perkalian	N *= 5, N akan dikali 5.
/=	Pembagian	N /= 5, N akan dibagi 5.
//=	Pembagian (dibulatkan ke bawah)	N //= 5, N akan dibagi 5 (dibulatkan ke bawah).
%=	Sisa Bagi / Modulo	N %= 5, N akan dimodulo 5.

1.4.3 Operator Relasional

Operator	Deskripsi	Contoh True	Contoh False
==	Sama dengan	2 == 2	2 == 3
!=	Tidak sama dengan	3 != 2	3 != 3
<	Kurang dari	2 < 3	2 < 2
>	Lebih dari	3 > 2	2 > 3
<=	Kurang dari sama dengan	2 <= 2	3 <= 1
>=	Lebih dari sama dengan	6 >= 5	2 >= 4

1.4.4 Operator Logika

Operator	Deskripsi	Contoh True	Contoh False
and	Dan: True jika kedua operand True	(1 < 2) and (3 == 3)	(1 == 2) and (3 == 3)
or	Atau: True jika salah satu operand True	(1 < 2) or (4 == 3)	(3 < 2) or (2 == 3)
not	Negasi: True jika operand False	not (3 < 2)	not (1 < 2)

1.5 Percabangan

Dalam pemrograman, terdapat percabangan. Dengan demikian, program kita dapat berperilaku tergantung input user. Misal kita buat program yang memeriksa apakah sebuah bilangan positif:

```
print("Masukkan nilai N: ", end="")
N = int(input())

if (N > 0):
    print(str(N) + " bilangan positif")

# str(N) berfungsi mengubah N menjadi string agar dapat dikonkatenasi saat
print
```

Lalu, jika kita ingin menuliskan kebalikannya:

```
...
if (N > 0):
    print(str(N) + " bilangan positif")
else: # N <= 0
```

```
...     print(str(N) + " bilangan bukan positif")
...
```

Namun, kita tahu kadang bilangan bisa nol atau negatif, jadi perlu kita tambahkan:

```
...
if (N > 0):
    print(str(N) + " bilangan positif")
elif (N < 0):
    print(str(N) + " bilangan negatif")
else: # N == 0
    print(str(N) + " bilangan nol")
...
```

Perhatikan juga kalau kita bisa membuat else ini berulang sampai yang kita mau. Selain itu, kita juga bisa meletakkan `if` di dalam `if`.

```
...
if (N >= 0):
    if (N > 0):
        print(str(N) + " bilangan positif")
    else: # N == 0
        print(str(N) + " bilangan nol")
else: # N < 0
    print(str(N) + " bilangan negatif")
...
```

1.6 Latihan Modul 1

1.6.1 Program Ganjil-Genap

Menggunakan contoh program bilangan positif, negatif, nol sebelumnya, kembangkanlah program tersebut sehingga ketika masukan bilangan positif program akan menentukan bilangan tersebut ganjil atau genap.

Contoh 1

Masukkan nilai N: <u>2</u> 2 bilangan positif genap
--

1.6.2 Soal Praktikum 1 Tahun 2022

Tuan Riz sedang melaksanakan ujian yang terdiri dari 14 soal isian singkat dan 2 soal esai. Ujian tersebut berlangsung selama 2 jam. Selama 1 jam pertama, Ia asik mengerjakan beberapa soal saja. Ia pun khawatir tidak dapat mengerjakan semua soal tepat waktu. Tuan Riz berencana untuk mengerjakan 1 soal isian singkat yang tersisa selama 10 menit dan 1 soal esai yang tersisa selama 20 menit.

Tentukan apakah Tuan Riz berhasil mengerjakan semua soal tepat pada waktunya.

Contoh 1

Masukkan banyak soal isian singkat yang sudah diselesaikan: <u>12</u> Masukkan banyak soal esai yang sudah diselesaikan: <u>1</u> Tuan Riz akan berhasil mengerjakan semua soal

1.6.3 Soal Praktikum 2 Tahun 2022

Tuan Leo menyebut sebuah bilangan sebagai bilangan "Bravo" jika jumlah dari setiap digitnya habis dibagi oleh setiap digitnya. Jika bilangan yang tidak memenuhi syarat tersebut disebut sebagai bilangan biasa dan disajikan bilangan 4 digit, Tentukan kategori bilangan tersebut.

Hint : Gunakan div dan mod untuk mendapatkan nilai dari masing-masing digitnya

Contoh 1

Masukkan sebuah bilangan: <u>4228</u> Bilangan tersebut adalah bilangan Bravo.

Penjelasan Contoh 1:

Jumlah dari setiap digit pada bilangan 4228 adalah $4 + 2 + 2 + 8 = 16$ dan 16 habis dibagi oleh setiap digitnya yaitu 2, 4, dan 8, sehingga disebut sebagai bilangan "Bravo".

Contoh 2

Masukkan sebuah bilangan: <u>3425</u> Bilangan tersebut adalah bilangan biasa.

1.6.4 Soal Praktikum 3 Tahun 2022

Seekor katak yang sedang diamati oleh Tuan Riz berada pada sebuah sumbu x . Katak tersebut melompat berdasarkan pola yang mudah. Lompatan pertama adalah a satuan ke kanan, lompatan kedua adalah b satuan ke kiri, lompatan ketiga adalah a satuan ke kanan, lompatan keempat adalah b satuan ke kiri, dan begitu seterusnya selama k kali. Bilangan k tersebut akan didapat berdasarkan posisi awal katak dengan aturan,

- Apabila katak berada pada koordinat yang genap, bilangan k -nya adalah kemampuan katak melompat ke kanan (a) dikali dengan kemampuan katak melompat ke kiri (b).
- Apabila katak berada pada koordinat yang ganjil, bilangan k -nya adalah $a \times 2$.
- Apabila katak berada pada koordinat negatif, kembali ke aturan pertama ($a \times b$).
- Akan tetapi, apabila katak berada pada koordinat negatif yang berkelipatan 3, bilangan k -nya adalah $b \times 2$.

Tentukan koordinat katak saat ini.

Contoh 1

Masukkan koordinat katak: 0
Masukkan panjang lompatan katak ke kanan: 5
Masukkan panjang lompatan katak ke kiri: 1
Koordinat katak sekarang adalah 13

Penjelasan Contoh 1:

Karena 0 termasuk pada aturan pertama, maka katak akan melompat sebanyak 5 kali. Perhitungan koordinat katak akan menjadi seperti ini: $0 + 5 - 1 + 5 - 1 + 5 = 13$

Contoh 2

Masukkan koordinat katak: -3
Masukkan panjang lompatan katak ke kanan: 2
Masukkan panjang lompatan katak ke kiri: 5
Koordinat katak sekarang adalah -18

2 Modul 2 - Perulangan dan Array

2.1 Pengulangan

Pada pemrograman, sering kali dibutuhkan pemrosesan berulang-ulang untuk mencapai suatu hasil tertentu. Apabila pengulangan ini dilakukan secara manual ukuran file program akan menjadi terlalu besar. Contoh sederhana adalah jika kita ingin menuliskan "Hello World" di layar sebanyak 1000 kali, maka akan dibutuhkan paling tidak 1000 baris perintah. Menggunakan sintaks pengulangan, persoalan tersebut dapat diselesaikan hanya menggunakan beberapa baris program.

```
for i in range(1000):  
    print("Hello World")
```

2.1.1 While Loop

Salah satu sintaks yang looping / pengulangan yang sering digunakan adalah sintaks While-Do. Program akan mengecek sebuah kondisi yang diberikan terlebih dahulu sebelum menjalankan statement yang ada di dalamnya.

Berikut adalah program yang menerima a dan b dan menuliskan $a, a + 1, a + 2, \dots, b - 1, b$.

```
a = int(input())  
b = int(input())  
i = a  
while (i <= b):  
    print(i)  
    i += 1
```

Perhatikan bahwa:

1. Sintaks while loop adalah **while (kondisi)**:
2. Perulangan body while loop akan terus dilakukan selama kondisi terpenuhi.
3. Body while adalah bagian kode yang indentasi ke dalam body while loop.

2.1.2 For Loop

Bentuk looping yang kedua adalah bentuk For. Bentuk ini umumnya digunakan untuk pengulangan yang sudah diketahui jumlahnya. Namun, for loop juga dapat dibuat menggunakan while-do loop.

Berikut adalah program yang menerima a dan b dan menuliskan $a, a + 1, a + 2, \dots, b - 1, b$.

```
a = int(input())  
b = int(input())  
for i in range(a, b+1):  
    print(i)
```

Perhatikan bahwa:

1. Sintaks for loop adalah **for variable in range(value)**:
2. Perulangan body for loop akan dilakukan sebanyak jumlah range yang diberikan.
3. Body for loop adalah bagian kode yang indentasi ke dalam body for loop.
4. Nilai variable akan berubah sesuai dengan nilai selanjutnya dalam range setiap perulangan.

`range` dapat digunakan dengan satu, dua, atau tiga parameter. Untuk lebih jelasnya, perhatikan potongan kode berikut:

```
range(10)          # 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
range(2, 5)         # 2, 3, 4
range(7, 2, -1)     # 7, 6, 5, 4, 3
range(2, 8, 2)      # 2, 4, 6
```

2.2 Perulangan bersarang

Perhatikan pula, perulangan dapat dilakukan di dalam perulangan. Sebagai contoh, berikut adalah program untuk membuat pola persegi.

```
n = int(input())
for i in range(n):
    for j in range(n):
        print("*", end="") # Dieksekusi sebanyak n * n kali
    print() # Dieksekusi sebanyak n kali
# Perhatikan indentasai di dalam perulangan.
```

Hal ini dapat dikembangkan bahwa body dari perulangan bersarang dapat berupa potongan kode lainnya. Kita dapat menggunakan I/O, operasi, dan perulangan lainnya dalam body loop. Kita dapat juga menggunakan while loop di dalam for loop dan sebaliknya.

2.3 Array

Array adalah variabel dengan satu buah nama, tetapi mengandung banyak nilai. Akses nilai-nilainya dilakukan dengan indeks.

Perhatikanlah contoh berikut!

Indeks	0	1	2	3	4	5	6	7	8	9
A	3	10	5	7	11	19	23	35	37	12

- `A[0] = 3`
- `A[1] = 10`
- `A[7] = 35`

Pada contoh diatas, kita memiliki sebuah variabel yang bernama A. Variabel A tersebut memiliki 10 buah nilai, dimana nilai-nilai tersebut dapat diakses dengan indeks. Untuk mengakses indeks

ke - x, gunakanlah A[x]. Dan nilai A[x] itu bisa kita anggap sebagai variabel yang berdiri sendiri. Konsep inilah yang kita sebut dengan array. Perhatikan pula bahwa indeks dimulai dari 0.

2.3.1 Deklarasi Array

Karena array juga merupakan sebuah variabel, maka array juga memerlukan deklarasi seperti variabel lainnya.

Contoh deklarasi array:

- `x = [0 for i in range(n)]` membuat array berukuran n dengan isi 0.
- `x = ["*" for i in range(100)]` membuat array berukuran 100 dengan isi karakter "*".

Untuk contoh tersebut array A yang terdefinisi adalah A[0], A[1], A[2], ..., A[9]. Mengakses nilai indeks di luar batasan tersebut akan menyebabkan runtime error. Oleh karena itu, tentukanlah rentang indeks yang akan digunakan saat deklarasi dengan tepat (sesuai dengan kebutuhan).

2.3.2 Array dan Variabel

Suatu array dapat kita anggap sebagai variabel, sehingga segala jenis operasi pada variabel juga berlaku pada array. Sebagai contoh, kita memiliki suatu array

```
tabel = [0 for i in range(10)]
```

Maka array tabel tersebut akan terdefinisi untuk indeks 0 sampai dengan indeks 9. Maka kita bisa melakukan instruksi

```
tabel[2] = int(input())
```

Jika diberikan 5 buah bilangan, dan kita perlu menyimpan bilangan tersebut pada tabel, kita bisa melakukan

```
tabel[0] = int(input("Masukkan nilai ke-0: "))
tabel[1] = int(input("Masukkan nilai ke-1: "))
tabel[2] = int(input("Masukkan nilai ke-2: "))
tabel[3] = int(input("Masukkan nilai ke-3: "))
tabel[4] = int(input("Masukkan nilai ke-4: "))
```

Namun, cara menginput tersebut kurang efisien. Akan lebih efisien jika kita menginput menggunakan perulangan (looping).

```
for i in range(5):
    tabel[i] = int(input('Masukkan nilai ke-' + str(i) + ': '))
```

2.3.3 String

Pada modul 1 kita telah mempelajari tipe data string. String dapat *dianggap* sebagai array of character, namun tipe data string memiliki karakteristik yang berbeda dengan array of character

yaitu string merupakan tipe data immutable. Artinya, string tidak dapat *diubah* nilainya setelah dideklarasikan. Sebagai contoh, jika kita memiliki string s1

```
s1 = "Hello"
```

Kita tidak dapat mengupdate nilai s1, misal melakukan Assignment s1[1] = 'a' akan menyebabkan error.

Berbeda halnya dengan array of character, kita dapat mengubah nilai array of character setelah dideklarasikan. Sebagai contoh, jika kita memiliki array of character s2

```
s2 = ['H', 'e', 'l', 'l', 'o']
```

Kita dapat mengubah nilai s2, misal melakukan assignment s2[1] = 'a' akan mengubah nilai s2 menjadi ['H', 'a', 'l', 'l', 'o'].

Namun, kita dapat mengakses nilai string dengan indeks, misal s1[1] akan bernilai 'e'.

2.4 Latihan Modul 2

2.4.1 Kelipatan 10 terkecil

Buatlah program yang menerima bilangan N dan menuliskan bilangan 10^x terkecil yang lebih dari N.

Contoh 1

```
Masukkan N: 5  
10
```

Contoh 2

```
Masukkan N: 1234  
10000
```

Loop manakah yang lebih baik digunakan untuk soal tersebut?

2.4.2 Segitiga Angka

Buatlah program yang menerima bilangan N dan membuat pola segitiga angka dengan dasar N seperti contoh.

Contoh 1

```
Masukkan N: 5  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

Loop manakah yang lebih baik digunakan untuk soal tersebut?

Dari kedua contoh tersebut, bagaimanakah perbedaan karakteristik kedua loop?

2.4.3 Reverse

Buatlah program yang menerima N buah bilangan dan menuliskan kembali bilangan-bilangan tersebut, namun dengan urutan terbalik.

Contoh

```
Masukkan N: 5  
5  
2  
1  
6  
3  
Hasil dibalik:  
3 6 1 2 5
```

2.4.4 Anagram

Diberikan dua array A dan B. Buatlah sebuah program yang dapat memeriksa apakah B merupakan anagram dari A. Asumsikan elemen pada array A dan B bilangan bulat [1,10].

Array B didefinisikan sebagai anagram dari array A bila elemen-elemen di array A dapat ditukar-tukar sehingga menghasilkan array B.

Hint: gunakan tabel frekuensi, yakni sebuah array yang indeks ke-1 menunjukkan banyaknya bilangan 1, indeks ke-2 menunjukkan banyaknya bilangan 2, dan seterusnya.

Contoh 1

```
Masukkan banyaknya elemen A: 3
Masukkan elemen A ke-1: 8
Masukkan elemen A ke-2: 4
Masukkan elemen A ke-3: 3
Masukkan banyaknya elemen B: 3
Masukkan elemen B ke-1: 4
Masukkan elemen B ke-2: 8
Masukkan elemen B ke-3: 3
B adalah anagram dari A
```

Contoh 2

```
Masukkan banyaknya elemen A: 2
Masukkan elemen A ke-1: 1
Masukkan elemen A ke-2: 4
Masukkan banyaknya elemen B: 2
Masukkan elemen B ke-1: 1
Masukkan elemen B ke-2: 5
B bukan anagram dari A
```

2.4.5 Palindrom

Buatlah program yang menerima sebuah string dan menuliskan apakah string tersebut palindrom. Asumsikan string hanya berisi huruf kecil (a-z).

Contoh 1

```
Masukkan panjang string: 5
Masukkan string: makan
makan bukan palindrom
```

Contoh 2

```
Masukkan panjang string: 3
Masukkan string: apa
apa adalah palindrom
```

Contoh 3

```
Masukkan panjang string: 10
Masukkan string: kasurruak
```

kasurrusak adalah palindrom

2.4.6 Soal Praktikum 1 Tahun 2021

Tuan Leo memiliki beberapa potongan lego dengan dimensi $1 \times 1 \times 1$. Tuan Leo ditugaskan membuat kubus sedikit mungkin dengan potongan-potongan lego yang ia miliki. Caranya adalah membuat kubus sebesar mungkin yang dapat dibuat dengan lego yang ada, lalu dengan sisa lego yang dimiliki, buat kubus sebesar mungkin yang dapat dibuat.

Tentukan berapa buah kubus yang dapat dibuat Tuan Leo.

Keterangan: Dilarang menggunakan built-in function untuk mencari akar dari sebuah bilangan. Penggunaan built-in function untuk mencari akar dari sebuah bilangan akan mengakibatkan nilai Anda pada soal ini 0.

Hint: Gunakan loop untuk mencari nilai kubik terdekat dengan jumlah lego yang dimiliki pada suatu waktu tertentu.

Contoh 1

Masukkan banyak potongan lego: 36
Tuan Leo dapat membuat 3 kubus.

Penjelasan contoh 1:

$$36 = 27 + 8 + 1$$

Contoh 2

Masukkan banyak potongan lego: 11
Tuan Leo dapat membuat 4 kubus.

Penjelasan contoh 2:

$$11 = 8 + 1 + 1 + 1$$

2.4.7 Soal Praktikum 2 Tahun 2021

Diberikan pertidaksamaan 2 variable sebagai berikut.

$$\left(\frac{x}{2}\right)^2 + \left(\frac{5y}{4} - 2\sqrt{|x|}\right)^2 \leq 120$$

Pada persoalan ini Tuan Dip meminta kalian untuk mengimplementasikan bentuk hati menggunakan * dengan bantuan pertidaksamaan tersebut pada rentang $-30 \leq x \leq 30$ dan $-15 \leq y \leq 15$. Jawaban anda dinyatakan benar jika bentuk hati dapat dilihat dengan jelas dan tidak terbalik maupun miring.

Notes: Gunakan fungsi $\text{abs}(x)$ untuk mengubah nilai x menjadi positif.

Contoh 1

[illegible]

2.4.8 Soal Praktikum 3 Tahun 2021

Buatlah program yang menerima bilangan asli $N > 1$ dan mengeluarkan faktor-faktor prima yang dimiliki bilangan tersebut. Hasil harus terurut mulai dari faktor prima terkecil hingga terbesar.

Contoh 1

Masukkan N: 110
Faktor primanya adalah 2, 5, 11.

Contoh 2

Masukkan N: 160
Faktor primanya adalah 2, 5.

2.4.9 Soal Praktikum 1 Tahun 2022

Tuan Leo diberikan sepotong list yang terdiri dari N elemen. Bantulah Tuan Leo untuk menentukan banyaknya potongan list (potongan list adalah hasil dari pemilihan beberapa indeks berurutan pada list) yang memiliki jumlah genap.

Contoh 1

```
Masukkan nilai N: 5
Masukkan bilangan ke-1: 1
Masukkan bilangan ke-2: 2
Masukkan bilangan ke-3: 3
Masukkan bilangan ke-4: 4
Masukkan bilangan ke-5: 5
Terdapat 6 potong list yang jumlahnya genap.
```

Penjelasan Contoh 1:

Potongan list yang jumlah elemennya genap adalah [1, 2, 3], [1, 2, 3, 4], [2], [2, 3, 4, 5], [3, 4, 5], dan [4]

2.4.10 Soal Praktikum 2 Tahun 2022

Tuan Leo memiliki kumpulan angka tidak terurut, tentukanlah berapa banyak grup angka terurut yang dapat dibentuk dari array tersebut. Data selalu distinct (nilai berbeda)

Nilai data yang dimasukkan dalam rentang 1-1000.

Contoh 1

```
Masukkan banyak data: 5
Masukkan data ke-1: 2
Masukkan data ke-2: 1
Masukkan data ke-3: 3
Masukkan data ke-4: 6
Masukkan data ke-5: 5
Banyak grup angka terurut adalah 2.
```

Penjelasan Contoh 1:

Grup terurut: [1,3] dan [5,6]

Contoh 2

```
Masukkan banyak data: 6
Masukkan data ke-1: 7
Masukkan data ke-2: 12
Masukkan data ke-3: 35
Masukkan data ke-4: 6
Masukkan data ke-5: 10
Masukkan data ke-6: 9
Banyak grup angka terurut adalah 4.
```

Penjelasan Contoh 2:

Grup terurut: [6,7], [9,10], [12,12], [35,35]

3 Modul 3 - Fungsi dan Prosedur

3.1 Fungsi

Apa itu fungsi? Fungsi adalah suatu bagian dari program yang mampu mengerjakan tugas atau operasi tertentu di luar program utama. Fungsi akan mengembalikan nilai sesuai algoritma yang diberikan.

Sebagai contoh, fungsi untuk menghitung kuadrat adalah sebagai berikut:

```
def kuadrat(x):  
    x2 = x * x  
    return x
```

Fungsi di atas bernama kuadrat dan mengembalikan nilai kuadrat. Fungsi itu juga menerima satu parameter bernama x.

Sebagai contoh, berikut program lengkap yang menerima input dan mengeluarkan kuadrat dari bilangan input.

```
def kuadrat(x):  
    x2 = x * x  
    return x2  
  
n = int(input())  
n2 = kuadrat(n)  
print(n2)
```

Perhatikan pada contoh di atas, variabel di program utama bernama n. Namun di fungsi, variabel berubah nama menjadi x. Meskipun x berubah, nilai n di program utama tidak akan berubah.

Sebuah fungsi juga dapat menerima lebih dari satu parameter. Selain itu, fungsi juga dapat melakukan hal-hal layaknya program biasa, namun tidak dapat mengubah variabel di program utama. Sebagai contoh, berikut adalah fungsi yang menghitung nilai a^b .

```
def pangkat(a, b):  
    # asumsi b >= 0  
    c = 1  
    for i in range(b):  
        c *= a  
    return c
```

Catatan: Anda tidak disarankan menaruh array sebagai parameter fungsi, karena ada hal khusus yang belum diajarkan di Berpikir Komputasional.

3.2 Prosedur

Prosedur sebenarnya sama seperti fungsi, namun tidak ada kembalian. Sebagai contoh, berikut adalah program untuk menuliskan menu:

```
print("Menu:")
```

```

print("1. Burger")
print("2. Ayam Geprek")
print("3. Mie Instan")
print("Masukkan pilihan: ")
pilihan_makanan = int(input())

print("Menu:")
print("1. Jus Alpukat")
print("2. Thai Tea")
print("3. Teh Tarik")
print("Masukkan pilihan: ")
pilihan_minuman = int(input())

print("Menu:")
print("1. Kentang")
print("2. Krupuk")
print("3. Abon")
print("Masukkan pilihan: ")
pilihan_tambahan = int(input())

```

Seperti yang tertulis di atas, menuliskan menu perlu berkali-kali. Kita dapat meringkasnya menjadi:

```

def tulis_menu(pil1, pil2, pil3):
    print("Menu:")
    print("1. " + str(pil1))
    print("2. " + str(pil2))
    print("3. " + str(pil3))
    print("Masukkan pilihan: ")

tulis_menu("Burger", "Ayam Geprek", "Mie Instan")
pilihan_makanan = int(input())

tulis_menu("Jus Alpukat", "Thai Tea", "Teh Tarik")
pilihan_minuman = int(input())

tulis_menu("Kentang", "Krupuk", "Abon")
pilihan_tambahan = int(input())

```

3.3 Latihan Modul 3

3.3.1 Fungsi Kuadrat

Anda memiliki sebuah fungsi berikut:

$$f(x) = x^2 - 2x + 5$$

Buatlah program yang menerima A dan B, lalu menuliskan semua nilai dari $f(A), f(A+1), \dots, f(B)$!

Catatan: Anda diwajibkan mengimplementasikan sebuah fungsi yang menerima parameter nilai x dan mengembalikan nilai $f(x)$.

Contoh

```
Masukkan A: 3
Masukkan B: 6
f(3) = 8
f(4) = 13
f(5) = 20
f(6) = 29
```

3.3.2 Segitiga Pascal

Segitiga pascal dapat dibuat menggunakan matriks. Matriks mulanya diinisialisasi dengan semua nilai baris pertama bernilai 1 dan semua nilai kolom pertama bernilai 1. Lalu, setiap isi matriks yang belum ada isinya diisi dengan jumlah dari bilangan di atas dan kirinya.

Buatlah sebuah program yang menerima masukan N dan menuliskan matriks segitiga pascal berukuran $N \times N$.

Contoh

```
Masukkan N: 4
1 1 1 1
1 2 3 4
1 3 6 10
1 4 10 20
```

3.3.3 Soal Praktikum 1 Tahun 2021

Tuan Ric dan Tuan Dip sering ke ITB bersama dengan kendaraan bermotor. Namun, Tuan Ric merasa bahwa cara membagi jadwal bayar bensin yang hanya bergantian merugikan dia. Maka Tuan Dip mengusulkan cara membagi jadwal bayar. Untuk setiap 2^n , mereka akan mengubah pola dengan menukar urutan (komplemen) pembayaran pada pola $2^{(n-1)}$. Berikut merupakan pola yang dimaksudkan Dip :

n	Pola
0	A
1	AR
2	ARRA
3	ARRARAAR

Dapat dilihat, teks yang dibold merupakan kebalikan dari bagian teks yang tidak bold pada suatu pola. Tuan Ric sangat menyukai ide ini. Bantulah Tuan Ric membuat sebuah program untuk menghasilkan pola tersebut.

Gunakanlah Fungsi atau Prosedur dalam pengerjaan soal ini. Clue : Sequence merupakan penambahan komplemen dari sequence sebelumnya seperti pada contoh. Siapkan array of character dengan ukuran 2 kali dari sequence sebelumnya.

Contoh 1

```
Masukkan karakter 1: A
Masukkan karakter 2: R
Masukkan nilai n: 2
Pola yang didapat: ARRA
```

Contoh 2

Masukkan karakter 1: U
Masukkan karakter 2: W
Masukkan nilai n: 4
Pola yang didapat: UWWUWUWWUWWUWWU