

UJIAN TENGAH SEMESTER KU1102/PENGENALAN KOMPUTASI SEM. 1 2022/2022 STREAM PEMROGRAMAN

HARI, TANGGAL : SABTU, 5 NOVEMBER 2022
PUKUL : 09.00 – 11.00 WIB
SIFAT UJIAN : INDIVIDUAL DAN CLOSED BOOK
DURASI : 120 MENIT
FAK./SEK. : FITB, FMIPA, FTTM, FTMD, FTSL, FTI, SITH, STEI

BACALAH PETUNJUK INI SEBELUM UJIAN DIMULAI :

- 1. JANGAN BUKA HALAMAN BERIKUTNYA SEBELUM UJIAN DIMULAI!**
- 2. Persiapan Ujian:**
 - a. Di atas meja hanya ada alat tulis dan identitas diri.
 - b. Ujian ini bersifat individual dan *closed book*.
- 3. SETELAH UJIAN DIMULAI:**
 - a. Cek apakah Anda sudah mendapatkan 7 lembar soal yaitu dari halaman 1 s.d.7. Jika Anda tidak mendapatkan lembar yang benar, tukarkan kepada pengawas.
 - b. Lembar soal adalah sekaligus lembar jawaban. Tidak disediakan lembar jawaban terpisah.
 - c. Tuliskan NIM, Nama, dan Kelas, pada tiap lembar soal pada tempat di pojok kanan atas.
 - d. Baca dulu semua soal berikut semua petunjuknya sebelum mengerjakan. Baca semua instruksi dengan sebaik-baiknya.
 - e. Kerjakan dulu soal yang Anda anggap mudah.
 - f. Tuliskan jawaban soal pada tempat-tempat yang disediakan sesuai instruksi soal. Jika Anda membutuhkan tambahan tempat untuk menjawab, gunakan halaman-halaman kosong yang ada pada lembar soal sesuai petunjuk dan jangan lupa menuliskan nomor soal dengan jelas. Jika Anda membutuhkan lembar corat-coret, gunakan juga halaman-halaman kosong yang tersisa pada lembar soal.
- 4. Tidak ada pertanyaan untuk semua soal. Buat asumsi sendiri jika ada soal yang menurut Anda tidak jelas dan tuliskan asumsi Anda bersama jawaban Anda.**
- 5. Sebelum memulai ujian, tanda tanganilah pernyataan di bawah ini.**

Dengan ini, saya menyatakan bahwa saya mengerjakan ujian ini secara jujur, tanpa bantuan yang tidak dibenarkan. Jika nantinya terbukti bahwa pernyataan saya ini tidak benar, maka saya bersedia menerima semua konsekuensinya sesuai aturan kuliah dan peraturan ITB.

Bandung, 5 November 2022

Nama:
NIM:

SELAMAT MENGERJAKAN. SEMOGA SUKSES. ☺

Bagian A. Computing System, Network, and Impact of Computing

Untuk soal 1 s.d. 3, tuliskan jawaban secara singkat, padat, tapi dengan jelas. Gunakan Bahasa Indonesia yang baik dan benar. Jangan lupa memberikan nomor soal dengan jelas.

Soal 1

Kecanduan *game* adalah salah satu problem remaja masa kini. Sebutkan (minimal 3) cara atau kiat untuk menghindari kecanduan *game*.

Soal 2

Dengan perkembangan teknologi dalam jaringan komputer, banyak aplikasi yang dikembangkan saat ini berbasis web. Sebutkan dua aplikasi berbasis web yang bisa digunakan untuk mendukung kerja dalam bidang keilmuan Fakultas/ Sekolah Anda, dan jelaskan dengan singkat bagaimana aplikasi tersebut mendukung kerja.

Soal 3

Suatu hari saat akan praktikum Pengenalan Komputasi, Anda menemukan komputer yang akan digunakan masih hidup dan pada browser, masih terlihat aplikasi media sosial terbuka. Anda mengenal pemilik akun yang sedang aktif pada aplikasi tersebut dan dia kebetulan adalah orang yang sangat tidak Anda sukai. Orang tersebut sering menggunakan akun media sosial ini untuk menyebarkan berita bohong (hoax). Anda berpikir, jika Anda melakukan sesuatu pada akun tersebut, misalnya mengirimkan pesan yang tidak pantas, menghapus file dan gambar, dan lain-lain., banyak orang yang akan merasa senang, bahkan berterima kasih pada Anda ☺. Lagipula salah dia sendiri tidak *log-out* dengan baik dari aplikasi tersebut di komputer umum. Apa yang akan Anda lakukan?

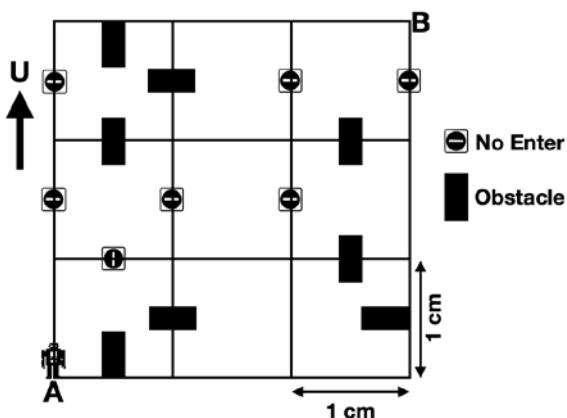
Tuliskan jawaban soal Bagian A di bawah ini atau di halaman kosong di balik hlm. 2 secara singkat, padat, tapi dengan argumen yang jelas. Gunakan Bahasa Indonesia yang baik dan benar. Jangan lupa memberikan nomor soal dengan jelas.

Bagian B. Berpikir Komputasional & Algoritmik

Soal 4

Sebuah robot mini bergerak dari titik A menuju titik B pada lintasan yang telah disediakan. Berikut adalah daftar perintah yang dapat dilakukan oleh robot.

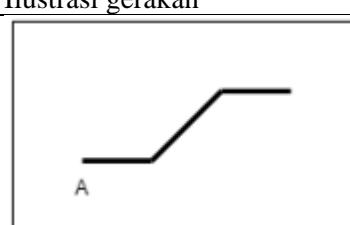
Perintah	Arti
Forward	Maju 1 langkah
TurnLeft d	Berputar/rotasi sebesar d derajat ke kiri. Asumsi $0 \leq d \leq 360$. Jika $d = 0$ atau 360 , maka robot tidak berubah arah.
TurnRight d	Berputar/rotasi sebesar d derajat ke kanan. Asumsi $0 \leq d \leq 360$. Jika $d = 0$ atau 360 , maka robot tidak berubah arah.
Repeat n (perintah)	Mengulang <i>perintah</i> sebanyak n kali. Asumsikan: $n \geq 0$. Jika $n = 0$, maka <i>perintah</i> tidak dilakukan sama sekali.



Beberapa informasi tambahan:

- Asumsi: Di awal, robot berada di posisi A dan robot menghadap ke **kanan/timur** (perhatikan gambar di atas).
- Lintasan yang dibentuk/dilewati oleh robot **hanya boleh** berada di dalam area denah saja.
- Jika robot harus melalui lintasan dengan penghalang/*obstacle*, maka robot harus melewatkinya dengan membentuk lintasan parabolik.
- Robot menerima rangkaian perintah yang dituliskan dalam urutan yang masing-masing dipisahkan sebuah koma (kecuali untuk perintah terakhir, lihat contoh di bawah). Rangkaian perintah dapat dituliskan menyamping, atau dapat juga dituliskan ke bawah, dan dapat juga dikombinasikan dengan spasi dan indentasi untuk memperjelas penulisan.
- Parameter *perintah* pada perintah **repeat** terdiri atas perintah-perintah yang bisa dijalankan oleh robot dan bisa terdiri atas lebih dari 1 perintah.

Contoh:

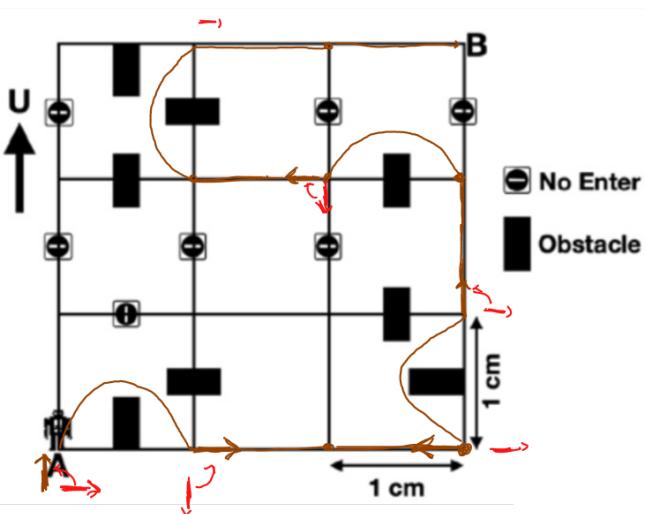
No.	Perintah	Arti	Ilustrasi gerakan
1	forward, turn L 45, forward, turn R 45 Alternatif penulisan: forward, turn L 45, forward, turn R 45	Pertama-tama robot maju 1 langkah (ke kanan), lalu berputar ke kiri 45 derajat, lalu maju 1 langkah, dan selanjutnya berputar ke kanan 45 derajat.	 <p>Catatan: setiap segmen garis diasumsikan sebesar 1 satuan gerakan</p>

2	<pre>repeat 180 (forward, turn L 1)</pre> <p>Alternatif penulisan:</p> <pre>repeat 180 (forward, turn L 1)</pre>	<p>Robot mengulang sebanyak 180 kali: maju 1 langkah dan berputar 1 derajat ke kiri.</p> <p>Ini akan membuat robot menggambar setengah lingkaran.</p>	
---	---	---	---

Tugas Anda:

- Gambarkan **lintasan terpendek** yang mungkin dilewati robot untuk begerak dari titik A ke B.
 - Tuliskan **urutan perintah** untuk membuat lintasan robot pada soal (a)

Tuliskan jawaban Bagian B di bawah ini atau di halaman kosong di balik hlm. 3 dan 4 (jangan lupa memberikan nomor soal dengan jelas).



L 90°, Repeat 180 (Forward, R 1), L 90°, Forward, Forward, L 180°,
Repeat 180 (Forward, R 1), L 90°, Forward, Repeat 180 (Forward, L 1),
R 90°, Forward, Repeat 180 (Forward, R 1), Forward, Forward

Bagian C. Algoritma dan Pemrograman

Untuk soal Bagian C:

- **Buatlah program** dalam bahasa pemrograman sesuai dengan yang diajarkan di kelas.
- **Bonus (hanya dinilai jika program dibuat, tidak perlu dibuat jika beresiko kehabisan waktu):**

Buatlah **flowchart/pseudocode** untuk persoalan tersebut (pilih salah satu saja sesuai yang ditekankan di kelas).

Soal 5

Spektrum optik adalah spektrum yang kontinyu sehingga tidak ada batas yang jelas antara satu warna dengan warna yang lainnya. Meskipun demikian, manusia dapat memberikan batas kira-kira untuk warna-warna spektrum yang berbeda-beda. Tabel berikut memberikan rentang panjang gelombang dari suatu warna yang dilihat oleh mata manusia normal. Data disesuaikan untuk kepentingan soal.

Warna	Panjang gelombang (dalam nm)
Ungu	380 - 450
Biru	451 - 495
Hijau	496 - 570
Kuning	571 - 590
Jingga	591 - 620
Merah	621 - 750

Buatlah sebuah program yang menerima masukan berupa panjang gelombang dalam nm (bertipe integer) dan menuliskan ke layar warna sesuai dengan panjang gelombangnya sesuai dengan deskripsi di atas. Jika masukan di luar rentang yang telah didefinisikan, dituliskan pesan kesalahan “Warna tidak terdefinisi”.

Contoh masukan dan keluaran:

Masukan Panjang Gelombang	Keluaran
495	Biru
250	Warna tidak terdefinisi

Tuliskan jawaban Soal 5 di bawah ini atau di halaman kosong di balik hlm. 5 (jangan lupa memberikan nomor soal dengan jelas).

```
Pjg Gel = int ( input (' --- ') )  
if ( Pjg Gel >= 380 ) and ( Pjg Gel <= 450 ) :  
    print ('Ungu')  
elif ( Pjg Gel >= 451 ) and ( Pjg Gel <= 495 ) :  
    print ('Biru')  
    :  
elif ( Pjg Gel >= 496 ) and ( Pjg Gel <= 570 ) :  
    print ('Hijau')  
elif ( Pjg Gel >= 571 ) and ( Pjg Gel <= 590 ) :  
    print ('Kuning')  
elif ( Pjg Gel >= 591 ) and ( Pjg Gel <= 620 ) :  
    print ('Jingga')  
elif ( Pjg Gel >= 621 ) and ( Pjg Gel <= 750 ) :  
    print ('Merah')  
else :  
    print ('Warna Tdk Terdefinisi')
```

Soal 6

Salah satu cara sederhana untuk memeriksa apakah sebuah bilangan bulat positif (> 1) merupakan bilangan prima atau bukan adalah dengan membagi bilangan tersebut dengan bilangan 1 hingga bilangan itu sendiri. Jika bilangan tersebut habis dibagi dengan bilangan selain 1 dan bilangan itu sendiri, maka bilangan tersebut bukan bilangan prima. Sebaliknya jika bilangan tersebut tidak habis dibagi dengan bilangan selain 1 dan bilangan itu sendiri, maka termasuk bilangan prima.

Buatlah program yang menerima input sebuah bilangan bulat positif (misal n, dengan asumsi $n > 1$), kemudian program memeriksa apakah bilangan tersebut prima atau bukan dengan menggunakan pengulangan pada program. Jika bilangan tersebut prima, maka ditampilkan ke layar bahwa bilangan tersebut prima; jika bukan maka ditampilkan ke layar bahwa bilangan tersebut bukan prima. Format tampilan pada layar bebas.

Contoh masukan dan keluaran:

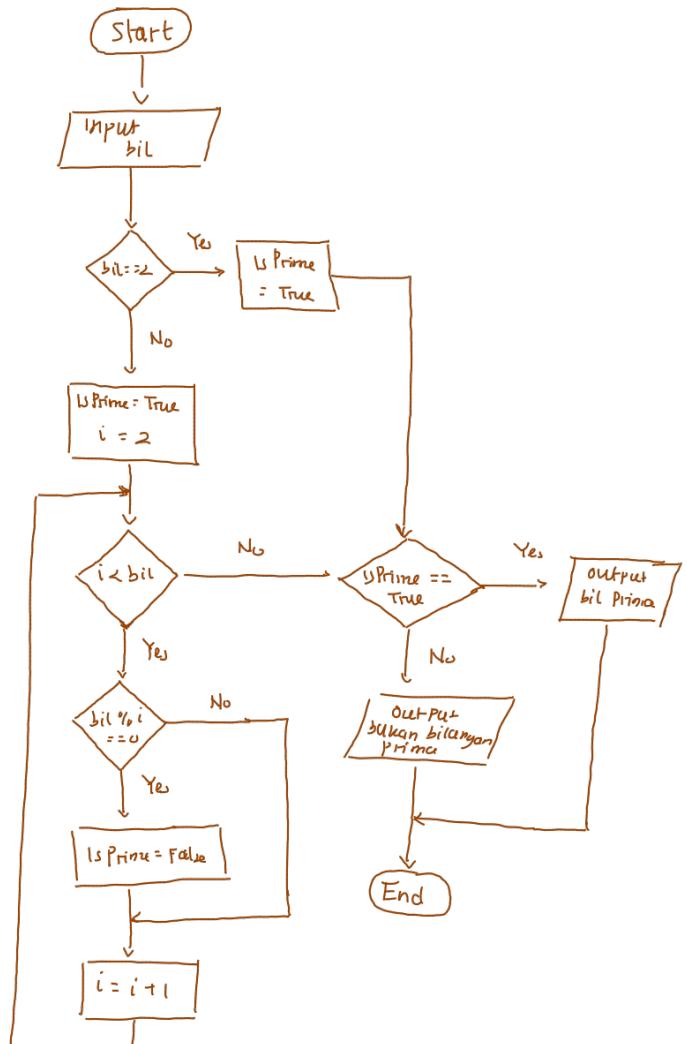
Masukan	Keluaran
4	4 bukan bilangan prima
2	2 adalah bilangan prima
307	307 adalah bilangan prima

$$\begin{array}{r} \textcircled{3} \rightarrow 7 \text{ } // \text{ } 2 \\ 2 \overline{)7} \\ \textcircled{1} \rightarrow 7 \% 2 \end{array}$$

Tuliskan jawaban Soal 6 di bawah ini atau di halaman kosong di balik hlm. 6 (jangan lupa memberikan nomor soal dengan jelas).

```

bil = int(input('---'))
if (bil == 2):
    isPrime = True
else:
    isPrime = True
    for i in range(2, bil):
        if (bil % i == 0):
            isPrime = False
    if (isPrime == True):
        print(f'Bilangan {bil} adalah prima')
    else:
        print(f'Bilangan {bil} bukan prima')
    
```



$$\begin{aligned}
 7 &\quad i \\
 7 \% 2 &= 1 \\
 7 \% 3 &= 1 \\
 7 \% 4 &= 3 \\
 7 \% 5 &= 2 \quad 4 \% 2 = 0 \Rightarrow \text{bukan prima} \\
 7 \% 6 &= 1
 \end{aligned}$$

Soal 7

Terdapat dua buah *array of integer* D1 dan D2. Himpunan domain nilai pada tiap elemen D1 dan D2 adalah {-1,1}. D1 adalah data hasil observasi, dan D2 adalah data hasil prediksi, dan keduanya memiliki jumlah elemen yang sama. Buatlah program yang meminta masukan dari pengguna banyaknya elemen array berupa bilangan bulat positif (misal N, asumsi N > 0), kemudian program akan meminta pengguna mengisi array D1 dan array D2 sejumlah N elemen, dengan setiap elemen pasti berisi -1 atau 1 (asumsi masukan pengguna selalu benar). Indeks elemen array mulai dari 0 (nol). Program akan menghitung:

- TP: banyaknya elemen D1 indeks ke-i yang bernilai 1 dan banyaknya elemen D2 indeks ke-i yang juga bernilai 1 ($D1[i] = D2[i] = 1$)
- TN: banyaknya elemen D1 indeks ke-i yang bernilai -1 dan banyaknya elemen D2 indeks ke-i yang juga bernilai -1 ($D1[i] = D2[i] = -1$)
- FP: banyaknya elemen D1 indeks ke-i yang bernilai -1 dan banyaknya elemen D2 indeks ke-i yang bernilai 1 ($D1[i] = -1$ dan $D2[i] = 1$)
- FN: banyaknya elemen D1 indeks ke-i yang bernilai 1 dan banyaknya elemen D2 indeks ke-i yang bernilai -1 ($D1[i] = 1$ dan $D2[i] = -1$)
- Nilai akurasi dengan formula: $Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$

Program akan menampilkan nilai TP, TN, FP, FN, dan akurasi ke layar. Contoh masukan dan keluaran:

Masukan	Keluaran
$N = 10$ $D1 = [1, -1, -1, 1, -1, 1, 1, -1, -1, 1]$ $D2 = [1, -1, 1, -1, 1, -1, 1, -1, -1, 1]$	$TP = 3$ ✓ $TN = 3$ ✓ $FP = 2$ ✗ $FN = 2$ ✗ $Akurasi = 0.6$

Catatan: elemen array yang dibandingkan antara D1 dan D2 selalu elemen dengan indeks yang sama.

Tuliskan jawaban Soal 7 di bawah ini atau di halaman kosong di balik hlm. 7 (jangan lupa memberikan nomor soal dengan jelas).

$N = \text{int}(\text{input}('---'))$

$D1 = [0 \text{ for } i \text{ in range}(N)]$

$D2 = [0 \text{ for } i \text{ in range}(N)]$

	$D1$	$D2$
0	0	0
1	0	0
2	0	0
3	0	0

$\text{for } i \text{ in range}(N) :$

$D1[i] = \text{int}(\text{input}('---'))$

$\text{for } i \text{ in range}(N) :$

$D2[i] = \text{int}(\text{input}('---'))$

$TP = 0$

$TN = 0$

$FP = 0$

$FN = 0$

```

for i in range (N) :
    if ( b1[i] == 1 ) and ( b2[i] == 1 ) :
        TP = TP + 1
    elif ( b1[i] == -1 ) and ( b2[i] == -1 ) :
        TN = TN + 1
    elif ( b1[i] == -1 ) and ( b2[i] == 1 ) :
        FP = FP + 1
    elif ( b1[i] == 1 ) and ( b2[i] == -1 ) :
        FN = FN + 1

```

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN)$$

print ('TP = {TP}')

print ('TN = {TN}')

print ('FP = {FP}')

print ('FN = {FN}')

print ('Akurasi= {Akurasi}')

- 2.a. Buatlah program untuk menghitung sisi miring sebuah segitiga siku-siku dengan rumus Pythagoras. Program membaca input 2 buah bilangan riil, misalnya a (alas segitiga) dan b (tinggi segitiga) dan menuliskan ke layar hasil perhitungan sisi miring c yang dihitung dengan rumus sbb.: $c = \sqrt{a^2 + b^2}$. Asumsikan $a > 0$ dan $b > 0$. Untuk menghitung x^2 gunakan $x ** 2$ dan untuk \sqrt{x} gunakan $x ** 0.5$.

```

a= float (input ('---'))
b= float (input ('--'))
c = ((a**2) + (b**2)) ** 0.5
print (f' --- {c} ')

```

- 2.b. Kelas pertandingan seorang atlit pertandingan bela diri ditentukan berat badannya sbb:

- $45 \text{ kg} \leq \text{berat badan} \leq 50 \text{ kg}$: kelas A
- $50 \text{ kg} < \text{berat badan} \leq 55 \text{ kg}$: kelas B
- $55 \text{ kg} < \text{berat badan} \leq 60 \text{ kg}$: kelas C
- $60 \text{ kg} < \text{berat badan} \leq 65 \text{ kg}$: kelas D
- $65 \text{ kg} < \text{berat badan} \leq 70 \text{ kg}$: kelas E
- berat badan $> 70 \text{ kg}$: kelas F

Buatlah sebuah program yang menerima berat badan seorang atlit (bilangan bulat) dan menuliskan kelas pertandingan atlit yang bersangkutan. Jika berat badan atlit $< 45 \text{ kg}$, tuliskan "Tidak memenuhi kualifikasi".

- 2.c. Diketahui sebuah deret sebagai berikut: $f(x) = 1 + x + x^2 + x^3 + \dots + x^n$. Buatlah program digunakan untuk menuliskan ke layar hasil $f(x)$ berdasarkan masukan nilai x (bilangan riil) dan n (bilangan bulat) dari pengguna dengan menggunakan deret di atas. Asumsikan: $-1 < x < 1$ dan $n > 1$. Untuk menghitung x^y , gunakan $x ** y$.

$x = \text{float} (\text{input} ())$	$\text{range}(6)$ 0 1 2 3 4 5	$\text{range}(2, 6)$ 2 3 4 5
$n = \text{int} (\text{input} ())$		
$ hasil = 0$		
$\text{for } i \text{ in range}(n+1) :$ $\quad hasil = hasil + (x ** i)$		
$\text{print} (f' --- {hasil})$		

- 2.d. Buatlah sebuah program yang menuliskan ke layar banyaknya bilangan yang merupakan bilangan bulat positif dan sekaligus genap dari sejumlah N buah bilangan yang tersimpan dalam *array of integer* T berukuran N. Tuliskan program Anda dalam *template* berikut. Baca instruksi yang tertera dalam komentar sebaik-baiknya.

```
# Program CountPosGenap
# Menghitung banyaknya bilangan positif dan genap dari sebuah array of integer

# KAMUS
# N : int
# T : array [0..N-1] of int - array of integer dengan indeks dari 0 s.d. N-1
# Tuliskan di bawah ini variabel lain yang diperlukan

# ALGORITMA
# Mendeklarasikan array dan mengisi array
N = int(input())
T = [0 for i in range(N)]
... # diasumsikan program untuk mengisi array T dengan sejumlah integer sudah ada,
# tidak diminta dibuat

# Tuliskan di bawah ini: program untuk menghitung ada berapa banyak bilangan integer yang
# positif dan genap dalam array T dan menampilkan hasilnya ke layar

Count = 0

for i in range(N) :
    if (T[i] > 0) and (T[i] % 2 == 0) :
        Count = Count + 1

print(f'--- {Count}' )
```

3.a. Tinggi Rata-Rata Siswa

Kepala Sekolah sebuah SMA ingin mendapatkan rata-rata tinggi badan siswa di suatu kelas. Untuk itu, dia memanggil semua siswa di kelas itu dan mengumpulkannya di ruang UKS untuk diukur tinggi badannya satu per satu. Karena tidak tahu banyaknya siswa dalam kelas, Kepala Sekolah berniat menghitungnya sambil mengukur tinggi badan siswa. Tinggi badan siswa diukur dalam satuan cm dan dengan pengamatan sekilas, bisa diketahui bahwa tidak ada siswa yang tingginya kurang dari 100 cm atau lebih dari 200 cm.

Seorang guru membantu Kepala Sekolah mengukur tinggi badan dan mencatatnya. Guru ini agak ceroboh sehingga sering memasukkan data yang salah. Data berat badan yang salah, yaitu kurang dari 100 cm atau lebih besar dari 200 cm, tidak digunakan untuk menghitung tinggi rata-rata.

Buatlah program yang membantu Kepala Sekolah untuk menghitung tinggi rata-rata siswa di kelas. Program menerima input sejumlah bilangan bulat yang mewakili tinggi mahasiswa. Pembacaan input diakhiri dengan memasukkan nilai -999 (tidak termasuk diproses). Selanjutnya program menuliskan tinggi rata-rata siswa di kelas berdasarkan data yang diinputkan, tanpa memperhitungkan data yang salah input (yaitu jika nilai < 100 cm atau > 200 cm). Jika tidak ada data yang bisa digunakan untuk menghitung tinggi rata-rata, tuliskan pesan "Tidak ada data".

Perhatikan contoh input/output di bawah ini (jumlah angka di belakang koma bebas):

Input	Output	Keterangan
120 145 150 -999	138.3333	
120 78 167 150 201 -999	145.6667	Nilai 78 dan 201 tidak dimasukkan dalam perhitungan karena 78 < 100 dan 201 > 200
-999	Tidak ada data	Pengguna langsung memasukkan -999 di awal
78 201 -999	Tidak ada data	Dari data yang dimasukkan tidak ada satu pun yang valid

Tuliskan jawaban soal 3.a di bawah ini atau di halaman kosong di balik hlm. 5.

```

Sum = 0
N = 0

Tinggi = Int (Input ('.' - '.'))

While (Tinggi != -999) :
    If (Tinggi >= 100) and (Tinggi <= 200) :
        Sum = Sum + Tinggi
        N = N + 1

    Tinggi = Int (Input ('.' - '.'))

    If (N == 0) :
        Print ('Tidak ada data')

    Else :
        Avg = Sum / N
        Print (f'.... {Avg}' )

```

3.b. Berat Badan Siswa

Di lain kesempatan, Kepala Sekolah ingin mendapatkan beberapa informasi terkait berat badan siswa karena beliau sangat peduli dengan kesehatan siswa. Tapi sekarang Kepala Sekolah sudah tahu bahwa ada 50 siswa di kelas. Untuk itu, dia menyiapkan daftar sejumlah 50 baris dan menyuruh seorang guru yang lain untuk mengukur dan mencatat berat badan siswa. Kali ini guru yang membantu Kepala Sekolah sangat kompeten. Dia tidak pernah salah memasukkan data berat badan siswa, yaitu berat seorang siswa pasti > 0 kg.

Buatlah program yang menerima masukan data berat badan 50 siswa (bilangan bulat, asumsikan masukan data berat tidak ada yang salah) dan:

- tampilkan nilai berat badan yang terbesar dari seluruh data berat badan siswa.
- baca sebuah nilai X dari masukan pengguna dan:
 - tuliskan berapa banyak siswa yang memiliki berat badan di atas X kg.
 - tampilkan data berat badan yang di atas X kg (ditampilkan hanya jika banyaknya siswa dengan berat badan di atas X kg > 0).

Contoh interaksi input/output:

Untuk meringkas tampilan, tidak semua data masukan/keluaran ditampilkan. Yang dicetak tebal dan digaris bawah adalah masukan dari pengguna.

Contoh-1
Masukkan data berat badan siswa =
<u>56</u>
<u>45</u>
<u>67</u>
<u>78</u>
<u>56</u>
...
<u>89</u>
<u>47</u>
<u>65</u>
<u>66</u>
<u>55</u>
Berat badan terbesar = 89
Masukkan berat badan = <u>70</u>
Banyaknya siswa dengan berat > 70 = 10
Daftar berat badan > 70 =
78
...
89

Contoh-2
Masukkan data berat badan siswa =
<u>36</u>
<u>44</u>
<u>73</u>
<u>78</u>
<u>46</u>
...
<u>69</u>
<u>47</u>
<u>65</u>
<u>66</u>
<u>76</u>
Berat badan terbesar = 78
Masukkan berat badan = <u>80</u>
Banyaknya siswa dengan berat > 80 = 0

Tuliskan jawaban soal 3.b di bawah ini atau di halaman kosong di balik hlm. 6.

maks = ✓ X ✓ ✓

```
BB = [0 for i in range(50)]
```

```
max = 0
```

```
for i in range(50):
```

```
    BB[i] = int(input('--'))
```

```
    if (BB[i] > max):
```

```
        max = BB[i]
```

```
print(f'----. {max}')
```

```
x = int(input('---!'))
```

```
N = 0
```

```
for i in range(50):
```

```
    if (BB[i] > x):
```

```
        N = N+1
```

```
print(f'----. {N}y')
```

```
If (N>0):
```

```
    print(f'daya --')
```

```
for i in range(50):
```

```
    if (BB[i] > x):
```

```
        print('BB[i]')
```

Soal 3.1. Volume Bola

Diketahui sebuah bola dengan jari-jari r . Volume bola adalah $V_{bola} = \frac{4}{3}\pi r^3$. Buatlah program yang menerima masukan jari-jari bola dan menampilkan volume bola. Untuk konstanta π , gunakan 3,1416.

asumsi jari ≠ merupakan bil. desimal

```
r = float (input('...'))
```

```
v = 4 * 3.1416 * (r ** 3) / 3
```

```
print ('Volume bola adalah {}')
```

Soal 3.2. Angka Armstrong

Angka Armstrong dari 3 (tiga) digit bilangan adalah bilangan bulat sehingga jumlah pangkat tiga dari masing-masing digitnya sama dengan angka itu sendiri. Misalnya, 371 adalah angka Armstrong karena $3^3 + 7^3 + 1^3 = 371$.

Buatlah program yang meminta user masukan bilangan bulat ratusan (3 digit), lalu menampilkan ke layar apakah bilangan tersebut adalah bilangan Armstrong atau bukan.

$$\begin{array}{r} 371 \\ 10 \overline{)371} \\ -30 \\ \hline 71 \\ -70 \\ \hline 1 \end{array} \rightarrow 371 \% 10$$

$$371 \% 10 = 1 \Rightarrow 1^3$$

$$371 \% 10 = 37$$

$$3 \% 10 = 3 \Rightarrow 1^3 + 7^3 + 3^3$$

```
Bil = int (input('...'))
```

$$37 \% 10 = 7$$

```
N = Bil
```

$$37 \% 10 = 3$$

```
hasil = 0
```

$$= \underline{\underline{}}$$

```
hasil = hasil + ((N \% 10) ** 3)
```

$$3 \% 10 = 0$$

```
K = N // 10
```

```
hasil = hasil + ((N \% 10) ** 3)
```

```
N = N // 10
```

```
hasil = hasil + ((N \% 10) ** 3)
```

```
If (hasil == Bil) :
```

```
    print ('Bilangan {} merupakan bil. Armstrong')
```

```
else :
```

```
    print ('Bilangan {} bukan merupakan bil. Armstrong')
```

371

$$1^3 + 7^3 + 3^3 = 371$$

```

Bil = int ( input ('...') )
N = Bil
hasil = 0

```

```

while (N > 0) :
    hasil = hasil + ((N % 10) ** 3)
    N = N // 10

if (hasil == Bil) :
    print ('Bulangan {}Bilangan merupakan bil. Armstrong')
else :
    print ('Bulangan {}Bilangan bukan merupakan bil. Armstrong')

```

Soal 3.3. Nilai Kuliah

Dosen ingin mengolah nilai kuliah yang dikuti oleh N orang mahasiswa ($N > 0$). Program tersebut menerima masukan nama, nilai praktikum, nilai uts dan nilai uas sebanyak jumlah mahasiswa (setiap nilai dalam skala 0 s.d. 100). Untuk setiap mahasiswa, dihitung nilai akhir dengan rumus sbb:

$$\text{nilai akhir} = 0,2 * \text{nilai praktikum} + 0,4 * \text{nilai uts} + 0,4 * \text{nilai uas}$$

Selanjutnya, berdasarkan nilai akhir tersebut, ditentukan indeks akhir mata kuliah yang didapat oleh setiap mahasiswa dengan aturan: $A \geq 85$, $85 > AB \geq 75$, $75 > B \geq 70$, $70 > BC \geq 60$, $60 > C \geq 50$, $50 > D \geq 40$, dan sisanya E.

Program kemudian mencetak nama, setiap komponen nilai, nilai akhir, dan indeks untuk tiap mahasiswa.

Berikut contoh tampilan outputnya (setiap detil nilai dipisahkan oleh spasi dan vertical bar, $N = 2$).

```

nama | praktikum | uts | uas | akhir | indeks
Udin | 50 | 50 | 50 | 50 | C
Asep | 90 | 70 | 70 | 80 | AB

```

Array

```
N = int ( input ('...') )
```

```
Nama = ['' for i in range(N)]
```

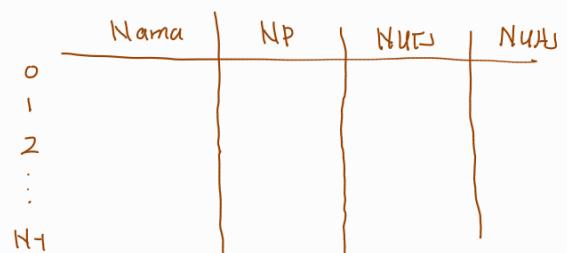
```
NP = [0 for i in range(N)]
```

```
NUTS = [0 for i in range(N)]
```

```
NUAS = [0 for i in range(N)]
```

```
NA = [0.0 for i in range(N)]
```

```
Idx = ['' for i in range(N)]
```



```

for i in range (N) :
    Nama [i] = input (' ... ')
    NP [i] = int (input ('... '))
    NUU [i] = int (input ('... '))
    NUAS [i] = int (input ('... '))

    NA [i] = (0.2 * NP[i]) + (0.4 * NUU[i]) + (0.4 * NUAS[i])

    if (NA[i] >= 85) :
        idx [i] = 'A'
    elif ((NA[i] >= 75) and (NA[i] < 85)) :
        idx [i] = 'AB'
    else :
        idx [i] = 'E'

print ('f' Nama | Praktikum | UTS ... )
for i in range (N) :
    print ('f' {Nama} | {NP[i]} | {NUU[i]} ... | {idx[i]} )

```

Soal 3.4. Deviasi Standar

Dalam statistik, deviasi standar adalah salah satu ukuran persebaran data. Deviasi standar untuk N buah sampel data dihitung dengan menggunakan rumus sbb.

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Keterangan: s = deviasi standar sampel

N = banyaknya data sampel

x_i = nilai data ke- i

\bar{x} = rata-rata nilai seluruh sampel data

$\sum_{i=1}^N P_i$ adalah penjumlahan setiap nilai P_i mulai dari $i = 1$ s.d. $i = N$

Suatu program menggunakan formula di atas untuk mengukur persebaran data tinggi mahasiswa di suatu universitas. Digunakan sampel data mahasiswa 1 kelas yang terdiri atas 50 mahasiswa. Program menerima masukan tinggi dari 50 mahasiswa (data tinggi adalah integer > 0) dan menyimpannya dalam suatu array of integer. Selanjutnya, dituliskan ke layar deviasi standar dari tinggi mahasiswa yang dihitung berdasarkan formula di atas.

Petunjuk: Sebelum memulai perhitungan deviasi standar, hitung terlebih dahulu rata-rata nilai seluruh sampel data \bar{x} .

$N = 50$

Tinggi = [0 for i in range(N)]

Jlh = 0

for i in range(N):

Tinggi[i] = int(input('...'))

Jlh = Jlh + Tinggi[i]

Rata = Jlh / N

S = 0

for i in range(N):

S = S + ((Tinggi[i] - Rata) ** 2)

S = S / (N-1)

S = S * 0.5

Print(f'Nilai deviasi standar adalah {sy}')

6. Jika diberikan $X = 6$, apakah hasil dari algoritma (dalam flowchart/pseudocode) berikut.

- a. 6
- b. 15
- ~~c. 21~~
- d. Tidak ada output

Flowchart:	Pseudocode:																
<pre> graph TD Start([Start]) --> Init[i = 0 Sum = 0] Init --> Input[/Input X/] Input --> Decision{i < X} Decision -- True --> Process[i = i + 1] Process --> Sum[Sum = Sum + i] Sum --> Decision Decision -- False --> Show[/Show Sum/] Show --> End([End]) </pre>	<pre> i ← 0 Sum ← 0 input (X) while (i < X) do i ← i + 1 Sum ← Sum + i { i ≥ X } output (Sum) </pre> <table border="0"> <tr> <td>i</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> <tr> <td>sum</td> <td>0</td> <td>1</td> <td>3</td> <td>6</td> <td>10</td> <td>15</td> <td><u>21</u></td> </tr> </table>	i	0	1	2	3	4	5	6	sum	0	1	3	6	10	15	<u>21</u>
i	0	1	2	3	4	5	6										
sum	0	1	3	6	10	15	<u>21</u>										

7. Algoritma pada soal 6 merepresentasikan persoalan:

- a. Penjumlahan X dengan i.
- ~~b. Penjumlahan X buah bilangan pertama dari 0.~~
- c. Penjumlahan i sebanyak X buah.
- d. Penjumlahan X sebanyak i kali.

8. Robot mini dapat menjalankan perintah berikut:

- V - satu langkah maju
- L (sudut) - rotasi ke sudut yang diberikan dalam kurung ke kiri,
- R (sudut) - rotasi ke sudut yang diberikan di dalam kurung kanan



Jika lebih dari satu perintah harus dieksekusi, perintah dapat dirangkai menggunakan operasi "+". Misalnya "V + L (20) + V + R (2)" berarti bahwa pertama-tama robot harus membuat langkah maju, lalu belok kiri 20 derajat, lalu buat langkah maju dan kemudian belok kanan 2 derajat.

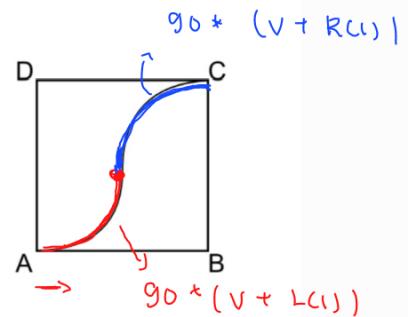
Untuk dapat mengulangi serangkaian perintah, dapat digunakan operasi "*". Misalnya "20 * (V)" berarti mengulangi 20 kali tindakan, yang diberikan dalam tanda kurung setelah "*", yaitu untuk membuat langkah maju. Jadi contoh ini akan menghasilkan 20 langkah ke depan.

Contoh lain: $180 * (V + L (1))$ menggambar setengah lingkaran.

Pada awalnya robot mini diposisikan pada titik A dan terlihat ke arah B.

Manakah dari kombinasi perintah di bawah ini yang telah ditetapkan yang akan membawa robot mini dari A ke C sepanjang jalur melengkung seperti pada gambar di samping?

- a. $90 * (V + L(1) + V + R(1))$
- b. $90 * (V + L(1)) + 90 * (V + R(1))$
- c. $90 * (V + L(1)) + R(30) + 90 * (V + R(1))$
- d. $L(90) + 90 * (V + L(1)) + R(90) + 90 * (V + R(1))$



Soal 2.B. Uraian

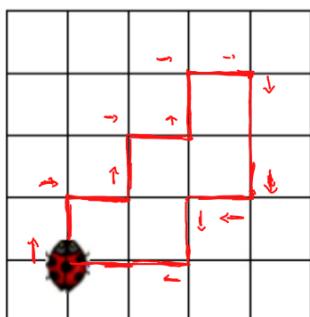
Ladybug (kepik) adalah robot dan dapat dikendalikan oleh perintah-perintah ini:

- **Forward N** - Kepik bergerak N langkah ($N \geq 1$) maju ke arah yang dituju.
 - **Left** - Kepik menghadap ke kiri tanpa bergerak ke depan.
 - **Right** - Kepik menghadap ke kanan tanpa bergerak ke depan.
 - **Repeat R (beberapa perintah)** - Ladybug mengulangi perintah dalam tanda kurung R kali jika $R \geq 1$. Jika $R < 1$, maka perintah dalam tanda kurung tidak dijalankan sama sekali.

Setiap gerakan maju dari Ladybug melukiskan sebuah trek (garis lurus) di lantai di mana ia bergerak.

1. Diberikan urutan perintah sebagai berikut kepada Ladybug (setiap perintah dipisahkan oleh koma):
Repeat 2 (Forward 1, Right, Forward 1, Left), Repeat 2 (Forward 1, Right),
Forward 2, Right, Forward 1, Left, Forward 1, Right, Forward 2, Right

Gambarkan trek yang dilalui oleh Ladybug yang dimulai pada posisi yang ditampilkan pada gambar berikut (gambar jawaban Anda langsung pada gambar di bawah ini dengan cara menebalkan garis yang menjadi trek).

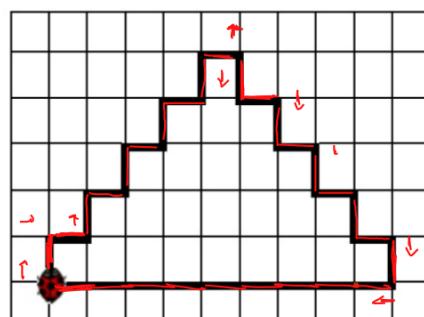
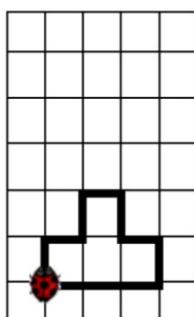
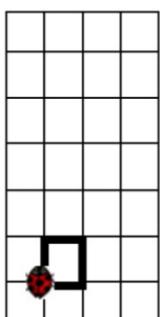


2. Diberikan X (bilangan bulat ≥ 1), tuliskan **urutan perintah** untuk membuat trek berbentuk piramida dengan ketinggian X dari posisi awal Ladybug (lihat contoh). Asumsikan selalu ada ruang yang cukup untuk menggambar piramida dari posisi mana pun. Perhatikan contoh-contoh berikut.

X = 1

X = 2

X = 5



Tuliskan jawaban untuk soal 2 di bawah ini atau di halaman kosong di balik halaman 5.

Repeat \times (Forward 1 , Right , Forward 1 , Left) , Repeat 2 (Right)

Repeat X-1 (Forward 1, Left, Forward 1, Right), Forward 1, Right

Forward $2x - 1$, right

```
Bil = Int ( Input ('...') )
```

```
N = Bil
```

```
hasil = 0
```

```
while (N > 0) :
```

```
    hasil = hasil + ((N % 10) ** 3)
```

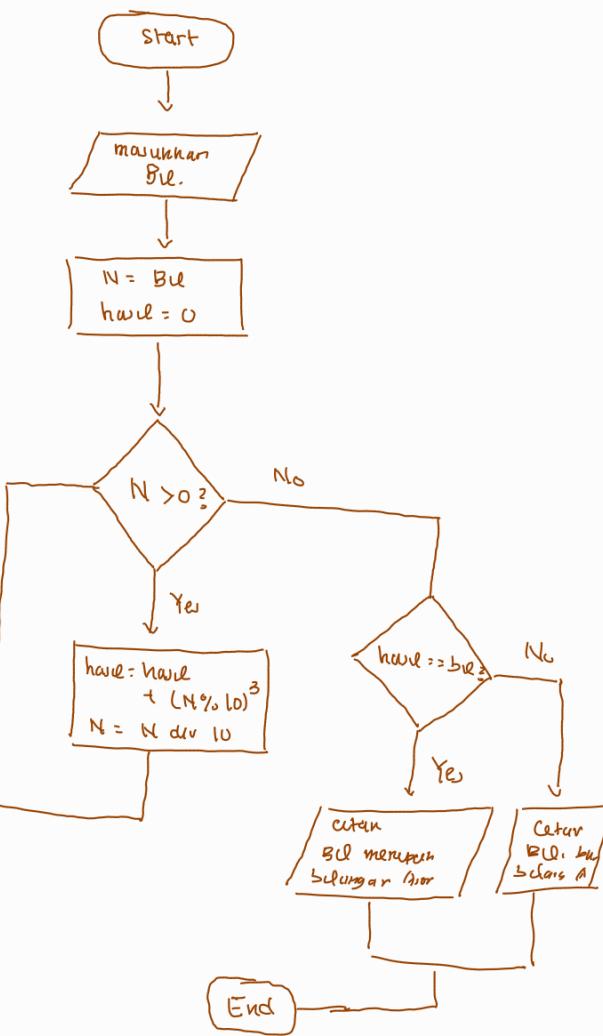
```
    N = N // 10
```

```
If (hasil == Bil) :
```

```
    Print (f' Bilangan {Bil} merupakan bil. Armstrong ')
```

```
else :
```

```
    Print (f' Bilangan {Bil} bukan merupakan bil. Armstrong ')
```



1. Peranti di bawah ini yang merupakan peranti computing adalah:
 - a. Elevator
 - b. Microwave ovens
 - c. Lemari pendingin
 - d. Smartphone
2. Di bawah ini yang bukan merupakan contoh hardware adalah:
 - a. Motherboard
 - b. CPU
 - c. RAM
 - d. Operating System
3. Windows, Linux, dan MacOS adalah contoh dari:
 - a. Operating system
 - b. Application
 - c. Browser
 - d. Hardware
4. Operasi dasar yang dilakukan oleh CPU adalah:
 - a. Menerima input dari pengguna
 - b. Memroses informasi
 - c. Menampilkan output
 - d. Menyimpan data
5. Perangkat utilitas digunakan untuk hal berikut, kecuali:
 - a. Kompresi file sehingga memakan tempat yang lebih kecil
 - b. Melakukan manipulasi aritmatika dan logik terhadap data
 - c. Mengkopi file antar media penyimpanan
 - d. Memperbaiki file yang rusak
6. Di bawah ini yang bukan contoh perangkat jaringan:
 - a. Antivirus
 - b. Network Interface Card (NIC)
 - c. Switch
 - d. Modem
7. Teknologi yang menghubungkan perangkat elektronik ke internet sehingga dapat berkomunikasi dan bertukar data disebut sebagai:
 - a. Internet
 - b. Artificial intelligence
 - c. Internet of things
 - d. Augmented reality
8. Berikut adalah keuntungan internet, kecuali:
 - a. Mengurangi waktu untuk mengembangkan program komputer
 - b. Saling berbagi sumber daya hardware komputer.
 - c. Berbagi data dan perangkat lunak.
 - d. Berkommunikasi dan bekerja bersama.