Prapraktikum 1

1. Praktikum 01/hello.c

   Buatlah program dalam Bahasa C dengan nama **hello.c** yang menuliskan ke layar:

   Hello, World!

   Penulisan diakhiri dengan enter. Perhatikan penulisan huruf besar, huruf kecil, spasi, tanda seru, dan tanda enter.

2. Praktikum 01/jumlah.c

   Buatlah program dalam Bahasa C yang mengimplementasikan fungsi penjumlahan 2 bilangan bulat

   Buatlah program fungsi penjumlahan dengan nama: jumlah.c

   Perhatian: terdapat newline setelah keluaran output.

3. Praktikum 01/main.c

   Pak Yoel merupakan dosen olahraga di ITB. Saat ini, beliau mempunyai stopwatch yang dapat mengukur lama waktu lari mahasiswa mengelilingi saraga. Sayangnya, stopwatch tersebut hanya dapat mengukur dalam waktu detik. Anda diminta untuk membuat kode program yang menerima masukkan detik dan mengeluarkan keluaran dalam format jam, menit, dan detik.

   Buatlah program konversi detik dengan nama: **main.c**

   Input:
   lama waktu dalam detik

   Output:
   konversi waktu dalam jam, menit, detik

   Contoh input:
   3820

   Contoh output:
   3820 detik = 1 jam 3 menit 40 detik

   Perhatian: terdapat newline setelah keluaran output.

Praktikum 1

1. Praktikum 01/bilangan.c

Bilangan subset genap adalah bilangan yang dapat dibentuk dari penjumlahan dua bilangan genap asli. Diberikan suatu bilangan, tentukan apakah bilangan tersebut merupakan bilangan subset genap!

Input:

Suatu bilangan dari 1 sampai 100

Output:

Suatu string Ya/Tidak

Contoh input:

8

Contoh output:

Ya

Penjelasan: bilangan 8 dapat dibentuk dari 4+4 sehingga 8 merupakan bilangan subset genap.

Perhatian: terdapat newline setelah keluaran output.

2. Praktikum 01/frekuensi.c

```
Diberikan sebuah string dengan panjang karakter maksimal 10 huruf. Tentukan banyaknya frekuensi angka 0-9 dalam string tersebut!

Input:
Sebuah string alfanumerik dengan panjang maksimal 10 huruf

Output:
Jumlah kemunculan angka 0-9 dalam string tersebut

Contoh Input:
a1123h56

Contoh Output:
0211011000

Penjelasan:
Dalam input
-Frekuensi kemunculan angka 0,4,7,8,9 adalah 0 kali
-Frekuensi kemunculan angka 2,3,4,5 adalah 1 kali
-Frekuensi kemunculan angka 1 adalah 2x

Note: terdapat newline setelah keluaran output
Hint: Nilai ASCII dari angka 0 adalah 48 dan praktikan dapat menggunakan fungsi strlen dari library string.h untuk menghitung panjang input karakter
```

3. Praktikum 01/strongnumber.c

Sebuah bilangan dapat disebut sebuah strong number apabila jumlah dari faktorial tiap-tiap digitnya sama dengan bilangan itu sendiri.

Buatlah sebuah program untuk menentukan apakah sebuah bilangan adalah strong number atau bukan.

Contoh:

Input	Output	Penjelasan
5	N	5! = 5*4*3*2*1 = 120 != 5
145	Y	1! + 4! + 5! = 1 + 24 + 120 = 145
111	N	1! + 1! + 1! = 3 != 111

Perhatian: terdapat newline setelah keluaran output.

Prapraktikum 2

1. Praktikum 02/cetakpersegi.c

Buatlah sebuah program bernama **cetakpersegi.c** yang mencetak sebuah persegi berdasarkan masukkan n, dimana panjang dari sisi persegi adalah 2n-1 dan dibentuk dengan pola O dan X secara selang-seling.

Contoh Input 1
1

Contoh Output 1
O

Contoh Input 2
2

Contoh Output 2
OXO
XOX
OXO

Contoh Input 3
3

Contoh Output 3
OXOXO
XOXOX
OXOXO
XOXOX
OXOXO

Perhatian: terdapat newline setelah keluaran output.

2. Praktikum 02/eq_string.c

Diberikan dua buah string yang terdiri dari huruf kapital dan nonkapital. Tentukan apakah kedua string tersebut sama! Kedua string dikatakan sama apabila memiliki panjang yang sama dan huruf per hurufnya merupakan abjad yang sama.

Buatlah program dengan nama: eq_string.c

Input:
Dua buah string, panjang string dijamin kurang dari 100

Output:
String Ya/Tidak

Contoh Input:
aNimeWasnTaMistake
animewasntamistake

Contoh Output:
Ya

Penjelasan:
Kedua string tersebut memiliki panjang dan abjad yang sama, sehingga merupakan string yang sama

Perhatian: terdapat newline setelah keluaran output

3. Praktikum 02/jumlah.c

Diberikan sebuah input integer. Buatlah sebuah program yang menjumlahkan semua digit bilangan dalam integer tersebut

Buatlah program dengan nama: jumlah.c

Input:
Sebuah Integer

Output:
Jumlah setiap digit dari integer tersebut

Contoh Input:
12465

Contoh Output:
18

Penjelasan:
1+2+4+6+5=18

Note: terdapat newline setelah keluaran output

Praktikum 2

1. Praktikum 02/reverse.c

Buatlah sebuah program yang menerima satu bilangan integer positif dan memberikan output integer tersebut dengan tiap digitnya sudah dibalik.

Contoh input:
12345

Contoh output:
54321

Terdapat newline pada akhir output.
Submit file **reverse.c**

2. Praktikum 02/indeksalstrukdat.c

Kelas IF2111 Algoritma dan Struktur Data baru saja menyelesaikan praktikum pertama. Setelah praktikum selesai, salah satu mahasiswa penasaran dengan banyaknya jumlah siswa yang lulus, indeks rata-rata serta indeks akhir kelas. Oleh karena itu, Ia membuat sebuah program untuk mencari informasi tersebut.

Program akan menerima input berupa **sebuah integer secara berulang hingga menerima input berupa -999**. Lalu, program akan mengeluarkan tiga baris kalimat berupa "Jumlah mahasiswa yang lulus = ", "Nilai rata-rata = ", dan "Indeks akhir kelas = ".

Adapun beberapa aturan yang diterapkan adalah:

1. Nilai yang valid dan masuk kedalam perhitungan adalah nilai yang berada diantara range 0-4. Nilai yang berada diluar range tersebut akan dianggap data kotor dan tidak dimasukkan kedalam perhitungan.
2. Aturan indeks yang diterapkan adalah sebagai berikut
    1. Apabila nilai = 4.00, Indeks = 'A'
    2. Apabila nilai >= 3.00 dan < 4.00, Indeks = 'B'
    3. Apabila nilai >= 2.00 dan < 3.00, Indeks = 'C'
    4. Apabila nilai >= 1.00 dan < 2.00, Indeks = 'D'
    5. Apabila nilai >= 0.00 dan < 1.00, Indeks = 'E'
3. Minimal indeks untuk lulus adalah B (>=3.00)
4. Nilai rata-rata ditulis dengan 2 angka desimal
5. Terdapat new line di akhir kalimat
6. Apabila tidak ada data yang valid (dari semua masukan, tidak ada yang memenuhi aturan nomor 1), maka program akan mengembalikan "Tidak ada data"

Berikut adalah contoh input-output dari program

Input	Output
1 2 3 4 -999	Jumlah mahasiswa yang lulus = 2 Nilai rata-rata = 2.50 Indeks akhir kelas = C
7 0 3 4 -999	Jumlah mahasiswa yang lulus = 2 Nilai rata-rata = 2.33 Indeks akhir kelas = C
4 4 4 4 -999	Jumlah mahasiswa yang lulus = 4 Nilai rata-rata = 4.00 Indeks akhir kelas = A
-999	Tidak ada data

3. Praktikum 02/bilangan.c

Sebuah bilangan dapat disebut bilangan cantik bila terdapat angka 7 di bilangan tersebut atau bisa dibagi 7.

Contoh:
- 14 adalah bilangan cantik karena dapat dibagi 7
- 71 adalah bilangan cantik karena terdapat angka 7 di puluhan

Buatlah program sederhana dengan nama **bilangan.c**.

Pertama-tama program menerima input bilangan **N**.

N menandakan jumlah test case. Untuk setiap test case, program menerima input bilangan M.

Untuk menyelesaikan setiap test case, hitunglah jumlah total bilangan cantik dalam rentang 1 sampai M.

Contoh input:
```
3
10
20
15
```

Contoh output:
```
7
38
21
```

Penjelasan:

Terdapat 3 test case.

Test Case	M	Hasil	Penjelasan
1	10	7	Bilangan cantik dari 1 sampai 10 cuma 7
2	20	38	Bilangan cantik dari 1 sampai 20 adalah: - 7 - 14 - 17 Jika dijumlahkan, hasilnya 38.
3	15	21	Bilangan cantik dari 1 sampai 15 adalah: - 7 - 14 Jika dijumlahkan, hasilnya 21

Perhatian:

Anda bisa langsung menampilkan hasil ketika menerima test case baru.

Tidak perlu menunggu sampai menerima semua input test case.

Contoh keadaan terminal yang benar:

- Jika langsung mengeluarkan output ketika menerima test case

```
3       // N jumlah test case
10      // Test case pertama
7       // Output test case pertama
20      // Test case kedua
38      // Output test case kedua
15      // Test case ketiga
21      // Output test case ketiga
```

- Jika menunggu menerima semua test case baru mengeluarkan output.

```
3       // N jumlah test case
10      // Test case pertama
20      // Test case kedua
15      // Test case ketiga
7       // Output test case pertama
38      // Output test case kedua
21      // Output test case ketiga
```

Prapraktikum 3

1. Praktikum 03/point.c

   Implementasikan point.h dan submit file dengan penamaan file bebas

   *Hint: Fungsi sqrt dapat digunakan dengan menggunakan library math.h

   **Note: Prosedur tulisPOINT mengeluarkan output dengan 2 desimal, TANPA diakhiri new line

2. Praktikum 03/panjangstring.c

   Buatlah implementasi dari file panjangstring.h, yaitu sebuah fungsi bernama `panjangString` yang menerima argumen pointer kepada karakter pertama dari sebuah string yang panjangnya tidak diketahui.

   String yang menjadi input disimpan sebagai array of char atau `char*` yang tersimpan secara kontigu.

   Implementasikan fungsi `panjangString` yang akan mengeluarkan panjang dari string tersebut.

   Dapat diasumsikan semua string input diakhiri dengan karakter spesial `'\0'` yang menandakan akhir dari string tersebut.

   Contoh:

String asli	Input fungsi	Return value	Penjelasan
hello	pointer yang menunjuk pada karakter pertama (h)	5	Array dalam memory: ['h', 'e', 'l', 'l', 'o', '\0']
four	pointer yang menunjuk pada karakter pertama (f)	4	['f', 'o', 'u', 'r', '\0']

   Submit file dengan penamaan bebas

3. Praktikum 03/arraydin.c

   Submit kembali file **arraydin.c** yang merupakan implementasi dari arraydin.h

Praktikum 3

1. Praktikum 03/garis.c

   Implementasikan garis.h dan submit file garis.c
   Sudah disediakan file point.h tapi tidak perlu diimplementasi.

   Anda dapat menggunakan fungsi-fungsi yang terdapat di **point.h** untuk membantu implementasi **garis.h**

2. Praktikum 03/jumlahkonsonanvokal.c

Buatlah implementasi prosedur **jumlahKonsonanVokal** dari file jumlahkonsonanvokal.h

jumlahKonsonanVokal merupakan sebuah prosedur yang menghitung jumlah huruf konsonan dan huruf vokal yang terdapat di suatu kalimat.

Input:
Semua karakter yang mungkin dalam suatu kalimat

Output:
Baris pertama merupakan jumlah konsonan
Baris kedua merupakan jumlah vokal

Contoh:

Input	Output
hello world	7 3
Object-oriented Programming	16 9
"Anime was a mistake", -Miyazaki Hayao.	14 15
July 4th is a special day for American.	18 12

Hint:
Buatlah fungsi tambahan **isAlphabet** dan **isVowel** untuk membantu implementasi prosedur **jumlahKonsonanVokal**.

3. Praktikum 03/arraydin.c

Submit kembali file **arraydin.c** yang merupakan implementasi dari arraydin.h

Prapraktikum 4

1. Praktikum 04/array.c

   Implementasikan array.h kedalam file bernama array.c

2. Praktikum 04/fibonacci.c

   Submit file **fibonacci.c** yang merupakan implementasi dari **fibonacci.h**

3. Praktikum 04/list.c

   Implementasikan list.h kedalam file bernama list.c

Praktikum 4

1. Praktikum 04/arrayMhs.c

   Buatlah ADT **arrayMhs.c** yang merupakan implementasi dari arrayMhs.h

2. Praktikum 04/arraymain.c

   Diberikan suatu array dengan ukuran N. Anggaplah isi dari array tersebut diulang tak hingga kali. Kemudian, diberikan rentang l dan r sebanyak q kali, hitunglah penjumlahan bilangan dari posisi l sampai posisi ke r!

   Batasan Input
   $1 \leq N \leq 10^3$
   $1 \leq q \leq 10^3$
   $1 \leq l \leq r \leq 10^3$

   Baris pertama N, menyatakan jumlah bilangan dalam array
   Baris kedua, berisi N buah bilangan dalam array
   Baris ketiga q, menyatakan jumlah *query*
   q baris selanjutnya berisi bilangan l & r, menyatakan rentang posisi

   Batasan Output
   Setiap output diakhiri dengan newline

   Contoh I/O:

Input	Output
4 1 2 3 4 2 1 2 1 10	3 23

   Penjelasan:
   Array [1, 2, 3, 4] jika diulang tak hingga kali menjadi: [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, ...]. Untuk rentang pertama {1, 2} berarti jumlahkan bilangan posisi 1 sampai posisi 2 (bukan index) sehingga 1+2 = 3. Untuk rentang kedua {1, 10} berarti jumlahkan bilangan posisi 1 sampai 10, jika dihitung didapatkan 23.

   Perhatian:
   Perlu dilakukan input & output pada program Anda.
   Penamaan file submission bebas.

3. Praktikum 04/MinMax.c

   Budi sedang bermain game dengan Andi. Dalam permainan tersebut, Budi harus bisa menghapus mencari nilai minimal dan maksimal dari sebuah list, menghapusnya dari urutan tersebut, kemudian menambahkan kedua nilai tersebut pada urutan terakhir list(nilai minimal ditambahkan terlebih dahulu dibandingkan nilai maksimal). Bantulah Budi membuat program tersebut dan submit filenya dengan nama MinMax.c

   contoh:

   Input:
   1 2 6 4 3

   Output:
   2 4 3 1 6

   Catatan:Mahasiswa hanya diminta untuk mengimplementasi MinMax.h. Buatlah fungsi yang mengembalikan List dan tidak perlu mencetak hasilnya. Gunakan list.h dan list.c untuk membantu menyelesaikan soal berikut

Prapraktikum 5

1. Praktikum 05/mesinkarakter.c

   Implementasikan mesinkarakter.h dan submit file mesinkarakter.c

2. Praktikum 05/hitungkarakter.c

   Dengan menggunakan implementasi ADT Mesin Karakter yang telah Anda buat di No 1, buatlah fungsi bernama **hitungkarakter** yang dapat menghitung jumlah kemunculan karakter pada suatu pita karakter yang didapatkan dari *stdin* dan mengembalikan nilai tersebut dengan spesifikasi fungsi sebagai berikut.

   int hitungkarakter(char karakter);

   Submit file dengan nama **hitungkarakter.c**

3. Praktikum 05/mesinkata.c

   Implementasikanlah **ADT Mesin Kata** dengan menggunakan implementasi **ADT Mesin Karakter V2** (bisa memakai yang telah dikerjakan pada nomor 1 dan diubah nama filenya).

   Diberikan file-file berikut:
   - mesinkata.h
   - boolean.h

   Submit file hasil implementasi **mesinkata.h**

Praktikum 5

1. Praktikum 05/hitungfrekuensi.c

   Diberikan sebuah kata. Buatlah fungsi hitungfrekuensi yang memiliki kegunaan untuk menghitung jumlah kemunculan kata 'tiga' pada kata tersebut. Kata 'tiga' dapat tersusun atas semua huruf lowercase, semua huruf uppercase ataupun campuran

   Input:
   ASDFtIgAFFFF.

   Output:
   1

   Implementasikan file hitungfrekuensi.h dan submit file dengan nama hitungfrekuensi.c. Gunakan mesinkarakter.h dan mesinkarakter.c untuk membantu menyelesaikan soal berikut

2. Praktikum 05/cipher.c

   Caesar cipher adalah sebuah metode enkripsi dimana setiap karakter dari teks awal/plaintext diganti dengan karakter lain dengan selisih tertentu dalam alfabet. Contohnya plaintext ABC menjadi BCD apabila dilakukan penggeseran sebanyak 1 huruf.

   Buatlah sebuah program **cipher.c** yang membaca sebuah pita karakter, lalu melakukan enkripsi dengan Caesar Cipher dan mencetaknya ke layar.

   Banyaknya penggeseran ditentukan dari panjang kata pertama, dan pita karakter tidak mungkin kosong.

   Masukan berupa karakter alfabet kapital A sampai Z (ASCII 65-90) dan keluaran diakhiri dengan newline. Gunakan boolean.h, mesinkarakterv2.h, mesinkarakterv2.c, mesinkata.h, dan mesinkata.c untuk membantu pengerjaan soal.

   Contoh masukan dan keluaran:

Masukan	Output	Keterangan
ANI PERGI KE PASAR DENGAN SEPEDA.	DQL SHUJL NH SDVDU GHQJDQ VHSHGD.	Banyaknya pergeseran = 3
X Y Z.	Y Z A.	Banyaknya pergeseran = 1

3. Praktikum 05/anagramalstrukdat.c

Kata yang bersifat anagram adalah kata yang tersusun atas huruf-huruf tepat sama seperti huruf penyusun kata alstrukdat. Contohnya kata yang bersifat anagram alstrukdat adalah datalstruk

Diberikan sebuah pita karakter, buatlah fungsi untuk menghitung jumlah kata yang bersifat anagram alstrukdat.

Input:
matkul favorit saya adalah alstrukdat sti karena alstrukdat memang seru meskipun ada yang menulis altrsukdat saya tetap suka.

Output:
3

Keterangan:
Yang terhitung sebagai anagram adalah: alstrukdat, alstrukdat, altrsukdat

Implementasikan fungsi **anagramalstrukdat.h** dan submit file bernama **anagramalstrukdat.c**
Silahkan gunakan mesinkarakterv2.h, mesinkarakterv2.c, mesinkata.h, dan mesinkata.c untuk membantu menyelesaikan soal berikut

Prapraktikum 6

1. Praktikum 06/queue.c

   Implementasikanlah ADT Queue dengan representasi array secara eksplisit dan alokasi statik.

   Diberikan file berikut:
   - **boolean.h**
   - **queue.h**

   Submit file hasil implementasi **queue.h**

2. Praktikum 06/circular_queue.c

   Implementasikanlah ADT Circular Queue dengan membuat file **circular_queue.c**!

   Gunakanlah file circular_queue.h dan boolean.h

3. Praktikum 06/reverse.c

   Implementasikan file header reverse.h, yang berisi sebuah fungsi transferReverse yang berfungsi memindahkan isi dari satu queue ke queue lain dengan urutan yang terbalik.

   Tidak diperbolehkan membuat array baru dalam pengerjaan soal ini. Gunakanlah fungsi-fungsi yang tersedia pada file circular_queue.h yang sudah kalian buat pada soal sebelumnya.

   Contoh:

q1 (Kondisi awal)	q2 (Kondisi akhir)
[1, 2, 3]	[3, 2, 1]
[1]	[1]

   Submit file **reverse.c**

Praktikum 6

1. Praktikum 06/tanpaBomb.c

   Implementasikan tanpaBomb.h kedalam file tanpaBomb.c

   Gunakan ADT queue yang telah dibuat pada pra-praktikum sebagai bantuan dalam menyelesaikan soal.

2. Praktikum 06/copyqueue.c

   Submit file **copyqueue.c** yang merupakan implementasi dari **copyqueue.h**. Gunakan bantuan fungsi yang sudah terimplementasi di **queue.c**.
   Implementasi hanya boleh menggunakan fungsi yang sudah disediakan (CreateQueue, isEmpty, isFull, enqueue, dll) dan makro(IDX_HEAD, IDX_TAIL, HEAD, TAIL).
   Tidak boleh mengubah atau mengakses Buffer, idxHead dan idxTail.

   copyqueue.h
   queue.h
   queue.c
   boolean.h

3. Praktikum 06/restoran.c

Ibu Gra adalah pemilik dari restoran bintag 5, Aldat, dan terkadang datang ke restoran tersebut untuk melihat kondisi di lapangan dan membantu memasak. Tidak seperti para koki yang lainnya, Ibu Gra hanya dapat memasak 1 masakan untuk 1 waktunya, dengan setiap masakannya membutuhkan waktu yang berbeda-beda, dan tamu-tamu lainnya dapat melakukan permintaan untuk mendapat masakan dari Ibu Gra, karena masakannya yang sangat populer dan juga viral di social media. Suatu hari, Ibu Gra ingin mengetahui berapa jumlah masakan yang telah disajikan, waktu minimal yang dibutuhkan, dan waktu maksimal yang dibutuhkan pada 1 hari tertentu. Menurut Ibu Gra, masalah ini dapat diselesaikan dengan membuat kode dalam bahasa C.

Spesifikasi program:

1. Input berupa angka, yaitu 0, 1, dan 2. 0 berarti program berhenti menerima masukan. 1 berarti menerima permintaan (dengan tambahan parameter adalah waktu yang dibutuhkan dan waktu yang tidak valid akan diabaikan). 2 berarti menyajikan masakan.\
2. Output berupa jumlah masakan yang disaji, waktu minimal yang dibutuhkan, dan waktu maksimal yang dibutuhkan (diikuti dengan newline). Default value adalah 0.
3. Nama file adalah **restoran.c**, dengan menggunakan header dari **queue.h**.

Berikut adalah contoh interaksi input/output program.

Input	Output	Keterangan
1 2 1 5 2 0	1 2 2	Karena hanya menyajikan 1 masakan, maka total adalah 1, waktu minimal dan waktu maksimal yang dibuutuhkan adalah 2
1 2 2 1 5 2 0	2 2 5	Karena menyajikan 2 masakan, maka total adalah 2, dan waktu minimal yang dibutuhkan adalah 2 (pada pesanan pertama) dan waktu maksimal yang dibutuhkan adalh 5 (pada pesanan kedua).

Prapraktikum 7

1. Praktikum 07/stack.c

   Buatlah program dalam Bahasa C yang mengimplementasikan fungsi pada file header stack.h

   Upload file *.c saja

2. Praktikum 07/palindrom.c

   Palindrom merupakan sebuah kata, bilangan, frasa, atau susunan karakter lain yang serupa jika dibaca dengan urutan terbalik ataupun tidak, seperti dalam kakak ataupun apa. Buatlah sebuah program bernama **palindrom.c** yang menerima input untuk memeriksa apakah sebuah stack merupakan stack yang palindrom atau tidak.

   Program akan menerima input berupa integer. Program akan berhenti menerima input ketika menerima masukan berupa 0. Setelah program berhenti menerima masukan, maka program akan mengeluarkan sebuah kata diantara "Palindrom", "Bukan Palindrom", atau "Stack kosong"

Input	Output	Keterangan
1 1 2 1 1 0	Palindrom	11211 dibaca dari kiri dan kanan sama-sama 11211
1 1 2 2 1 0	Bukan Palindrom	11221 dibaca dari kanan menjadi 12211
1 0	Palindrom	1 dibaca dari kiri dan kanan sama-sama 1
0	Stack kosong	Stack tidak berisi apa-apa

   Note:
   - perhatikan bahwa terdapat newline di akhir keluaran
   - panjang input tidak akan melebihi MaxEl

3. Praktikum 07/aritmatika.c

   Submit file **aritmatika.c** yang merupakan implementasi dari **aritmatika.h**. Gunakan implementasi stack dari nomor 1 pra-praktikum Diberikan ekspresi aritmatika yang ditulis dalam bentuk postfix, evaluasi ekspresi tersebut sehingga menghasilkan suatu nilai.

Praktikum 7

1. Praktikum 07/browserhistory.c

   Implementasikan browserhistory.h dengan membuat browserhistory.c

   Perhatikan infotype pada ADT stack adalah string (char*).

2. Praktikum 07/minStack.c

Buatlah program yang membaca 2 (dua) buah string, S1 dan S2, yang masing-masing merepresentasikan sebuah integer besar (big integer) positif atau 0. Panjang setiap string maksimum adalah 100, sehingga integer maksimum yang bisa direpresentasikan terdiri atas 100 digit. Panjang string minimum adalah 1, yaitu jika integer yang direpresentasikan hanya terdiri atas sebuah digit. Selanjutnya program akan menampilkan hasil pengurangan kedua integer tersebut (S1 - S2).

Petunjuk:

- Gunakan stack untuk mensimulasikan pengurangan tiap digit dari integer. Pakailah ADT Stack (file header stack.h) yang telah Anda kerjakan sebagai tugas pra-praktikum.
- Konversi masing-masing karakter dalam string menjadi sebuah integer dan dengan menggunakan stack of integer, push setiap integer ke dalam stack. Dengan demikian, bilangan yang merepresentasikan satuan (digit terakhir bilangan) akan berada di top stack.
- S1 mungkin kurang dari S2. Oleh karena itu perlu dilakukan pemeriksaan apakah S1 kurang dari S2, jika iya lakukan pengurangan dengan cara S2-S1 kemudian tandai bahwa nilainya negatif.

Contoh I/O:

Input	Output
16 16	0
0 12345	-12345

3. Praktikum 07/valid.c

Submit file **valid.c** yang merupakan implementasi dari valid.h. Gunakan implementansi stack dari nomor 1 pra-praktikum, dengan infotype di sini adalah **char**, bukan **int** (Perhatikan!)

Prapraktikum 8

1. Praktikum 08/set.c

   Buatlah program yang mengimplementasikan fungsi pada file header set.h

   Upload file **set.c**

2. Praktikum 08/map.c

   Buatlah program yang mengimplementasikan fungsi pada file header map.h.

   Upload file map.c

3. Praktikum 08/hashmap.c

   Buatlah program yang mengimplementasikan fungsi pada file header hashmap.h.

   Upload file hashmap.c

Praktikum 8

1. Praktikum 08/duplicate.c

   Implementasikanlah fungsi-fungsi pada duplicate.h
   Diberikan file berikut:
   - set.h
   - set.c
   - duplicate.h

   Submitlah kembali file yang telah berisi implementasi fungsi-fungsi pada duplicate.h

2. Praktikum 08/union_map.c

   implementasikan fungsi **UnionMap** yang pada pada file union_map.h . Submit file dengan nama **union_map.c**!

   Untuk membantu pekerjaan, gunakan file map.h dan map.c berikut

3. Praktikum 08/set.c

   Gunakan hasil pra-praktikum kalian untuk mengimplementasikan set.h, header untuk ADT Set dengan beberapa fungsi tambahan untuk melakukan operasi terhadap ADT Set.

Prapraktikum 9

1. Praktikum 09/listlinier.c

   Buatlah implementasi listlinier.c dari file header listlinier.h!

2. Praktikum 09/driiverlistlinier.c

   Dengan menggunakan ADT List Linier dengan representasi fisik pointer yang Anda buat pada soal 1, buatlah program yang mengelola statistik daftar nilai sebuah mata kuliah (asumsikan nilai selalu 0..100) sebagai berikut:

   1. Membuat sebuah list kosong untuk menyimpan daftar nilai mata kuliah.
   2. Untuk setiap list: Membaca sekumpulan nilai integer dari keyboard sampai pengguna memasukkan angka di luar range nilai (di luar 0..100). Setiap elemen ditambahkan sebagai elemen pertama.
   3. Jika list tidak kosong, menuliskan:
      - Ada berapa banyak nilai yang ada di daftar
      - Nilai tertinggi yang ada di daftar.
      - Nilai terendah yang ada di daftar.
      - Nilai rata-rata (presisi: 2 digit di belakang koma).
      - Mengkopi isi daftar ke sebuah list yang lain dan membalik isinya dan menuliskan isinya.
      - Menuliskan isi daftar yang asli (sebelum dibalik).
   4. Jika list kosong, menuliskan "Daftar nilai kosong".

   Berikut adalah contoh input/output:

Input	Output
100 98 78 100 76 89 -1	6 100 76 90.17 [100,98,78,100,76,89] [89,76,100,78,98,100]
-999	Daftar nilai kosong

3. Praktikum 09/OddEvenList.c

   Implementasikan fungsi OddEvenList yang pada pada file OddEvenList.h . Submit file dengan nama **OddEvenList.c**!

   Gunakan ADT list linier yang dibuat pada nomor 1.

Praktikum 9

1. Praktikum 09/cache.c

Pada sistem operasi, akses memori pada RAM cukup lambat. Karena itu, digunakan cache yang jauh lebih cepat. Namun cache berukuran sangat kecil, jadi cache hanya menyimpan sedikit bagian dari RAM. Karena tidak semua nilai disimpan, saat dibutuhkan sebuah nilai x, cache bisa hit (cache memiliki nilai x) atau miss (cache tidak memiliki nilai x).

Salah satu implementasi cache adalah dengan skema LRU, yakni Least Recently Used. Pada skema ini, apabila nilai x tidak ada di cache, cache akan menghapus nilai di cache yang paling lama sudah tidak digunakan. Jadi, cache menyimpan daftar nilai, terurut dari yang paling baru digunakan, sampai yang paling lama digunakan. Cache dapat direpresentasikan sebagai sebuah list linier. Cache akan diinisialisasi dengan nilai 1 sampai N. Lalu, akan ada Q buah operasi pengambilan nilai x:

- Jika x ada di cache, nilai x dipindah ke depan cache.
- Jika x tidak ada di cache, nilai paling akhir dihapus dari cache dan x dimasukkan ke depan cache.
- Untuk setiap operasi, tuliskan apakah operasi "hit" atau "miss". Lalu, tuliskan isi list.

Gunakan listlinier.h, listliner.c, dan kumpulkan **cache.c**

Contoh input / output:

Input	Output	Penjelasan
6 5 4 6 7 5 5	hit [4,1,2,3,5,6] hit [6,4,1,2,3,5] miss [7,6,4,1,2,3] miss [5,7,6,4,1,2] hit [5,7,6,4,1,2]	N = 6 Q = 5 Pada awalnya, cache berisi [1,2,3,4,5,6]. Ada 5 operasi yang dilakukan. Pada operasi pertama, 4 ada di cache. Pada operasi kedua, 6 juga ada di cache. Pada operasi ketiga, 7 tidak ada di cache, jadi 5 dihapus dan 7 ditambahkan ke depan cache. Pada operasi keempat, 5 sudah tidak ada di cache, jadi 3 dihapus dan 5 ditambahkan ke depan cache. Pada operasi kelima, 5 sudah ada di cache.

2. Praktikum 09/listfibonacci.c

List fibonacci adalah list yang elemennya merupakan jumlah dari 2 elemen sebelumnya. Pada awal program, program akan meminta jumlah elemen dari list fibonaci yang akan dibuat. Selanjutnya, program akan mengeluarkan output sesuai aturan berikut:

- Apabila jumlah elemen = 0, maka program akan langsung mengembalikan list kosong.
- Apabila jumlah elemen = 1, maka program akan meminta sebuah inputan yang menjadi elemen dalam list
- Apabila jumlah elemen >=2, maka program akan meminta dua buah inputan. Inputan pertama adalah elemen pertama dalam list dan inputan kedua adalah elemen kedua dalam list

Berikut adalah contoh input/output dari program

Input	Output
0	[]
1 2	[2]
1 4	[4]
2 0 1	[0,1]
4 3 4	[3,4,7,11]

Untuk mempermudah pengerjaan, silahkan lengkapi file template.c (boleh mengganti blok kode program yang telah ditulis pada template) dan submit dengan format nama listfibonacci.c

Note: Semua input berupa bilangan bulat positif

Diberikan ADT yang digunakan adalah:
- listlinier.h
- listlinier.c

3. Praktikum 09/removeDuplicate.c

Buatlah implementasi **removeDuplicate.c** dari file header removeDuplicate.h!
Perhatian:
- List masukan sudah **terurut membesar**
- List hasil juga harus **terurut membesar**

- removeDuplicate.h
- listlinier.h
- listlinier.c
- boolean.h

Prapraktikum 10

1. Praktikum 10/listdp.c

   Implementasikan file listdp.h! Submit dengan nama file **listdp.c**

2. Praktikum 10/listsirkuler.c

   Submit file **listsirkuler.c** yang mengimplementasikan listsirkuler.h

3. Praktikum 10/elemenKeN.c

   Buatlah sebuah program yang dapat mengetahui elemen ke-N dari sebuah list menggunakan sebuah fungsi perantara ElemenKeN.

   Program utama akan membaca input seperti pada tabel berikut

Input	Output	Penjelasan
0 0	List Kosong	List = [] N=0
0 1	List Kosong	List = [] N=1
1 2 3 4 5 0 0	1	List = [1,2,3,4,5] N=0 L(0) = 1
1 2 3 4 5 0 2	3	List = [1,2,3,4,5] N=2 L(2) = 3
1 2 3 4 5 0 4	5	List = [1,2,3,4,5] N=4 L(4) = 5
1 2 3 4 5 0 6	2	List = [1,2,3,4,5] N=6 L(6) = 2
1 2 3 4 5 0 8	4	List = [1,2,3,4,5] N=8 L(8) = 4

   Program akan menerima input yang kemudian akan dimasukkan kedalam list hingga menerima input berupa angka 0. Setelah menerima angka 0, maka program akan meminta lagi sebuah inputan sebagai N yang akan dicari nilai elemennya.

   Untuk mempermudah pengerjaan, silahkan lengkapi file template.c dan submit dengan format nama elemenken.c

   Note: Semua input berupa bilangan asli positif

   Gunakan ADT listsirkuler.c dan listsirkuler.h yang telah dibuat pada soal sebelumnya

Praktikum 10

1. Praktikum 10/linkstack.c

   ADT Stack dapat direpresentasikan dengan list linier. Implementasikan linkstack.h dengan membuat linkstack.c!

2. Praktikum 10/queuelist.c

   Representasi Queue dengan List Linier

   Nama File: **queuelist.c**

   Buatlah program body dalam Bahasa C yang mengimplementasikan fungsi pada file header **queuelist.h**

   Upload file **queuelist.c** saja.
   - queuelist.h
   - boolean.h

3. Praktikum 10/linkdummy.c

   Implementasikan linkdummy.h dengan membuat linkdummy.c!

   Catatan:

   - Elemen dummy adalah node dengan nilai infotype 0