

TUGAS PENDAHULUAN

MODUL X

Stack

Pada modul ini, *library* yang digunakan **hanya** `stdio.h`. Apabila membutuhkan fungsi matematika pada `library math.h`, maka buatlah fungsi tersebut menjadi **fungsi/prosedur baru**.

Pengenalan

Pengertian Stack

Stack adalah struktur data yang berbentuk tumpukan. Data yang kita pakai dapat dibayangkan sebagai tumpukan piring.



Gambar 1: Ilustrasi stack dengan piring

Satu piring mewakili satu buah data. Karena piring-piring itu berada dalam tumpukan, piring yang dapat diambil hanyalah piring yang berada di paling atas. Demikian juga apabila kita ingin menambah piring baru, hanya bisa kita letakkan di atas tumpukan.

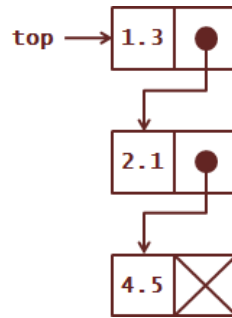
Sekarang kembali ke level abstraksi pemrograman, **stack** adalah tumpukan data. Menambahkan data hanya bisa dilakukan ke atas stack. Proses menambahkan data pada stack disebut **push**. Mengeluarkan data juga hanya bisa dilakukan untuk data yang berada di paling atas stack. Proses pengeluaran data disebut **pop**. Sebagai tambahan, data di dalam stack juga biasa disebut sebagai **element**.

Pada tugas pendahuluan ini, Anda diminta untuk mengimplementasikan tipe data bentukan untuk **stack**, fungsi **push()**, dan fungsi **pop()**. Ketiga-tiganya akan digunakan di dalam praktikum.

Problem 1 : Implementasi Struktur Data Element Stack

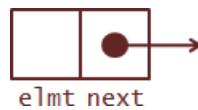
Definisi Masalah

Pada praktikum yang akan dilaksanakan, struktur data stack diimplementasikan sebagai sebuah linked-list. Data yang akan kita gunakan sebagai “piring” dalam stack bertipe double.



Gambar 2: Ilustrasi stack berisi 3 element double diimplementasikan dengan *linked-list*.

Untuk mempermudah pekerjaan kita ke depan, kita harus membuat tipe bentukan untuk **satu element dari stack**.



Gambar 3: Satu element (*node*) stack.

Tipe bentukan untuk satu element stack ini, kita sebut sebagai **node**. Data di dalamnya diberi nama **elmt** dan bertipe double. Pointer ke node selanjutnya diberi nama **next**. Implementasikan tipe bentukan untuk *node*.

Diberikan kode sebagai berikut:

```
typedef struct node
{
    /** Isi apa yang kurang
    Pada struktur ini
    ...
    ... **/
}node;
```

Problem 2 : Implementasi Struktur Data Stack

Definisi Masalah

Setelah kita mengimplmentasikan tipe untuk satu element atau *node* dari stack, selanjutnya adalah membentuk tipe untuk **stack** itu sendiri. Ilustrasinya dapat dilihat pada gambar 2 diatas. Pointer yang menunjuk ke element paling atas dari stack di beri nama **top**.

Diberikan bantuan kode sebagai berikut:

```
typedef struct stacktype
{
    /** Isi apa yang kurang
    Pada struktur ini
    ...
    ... **/
}stacktype;
```

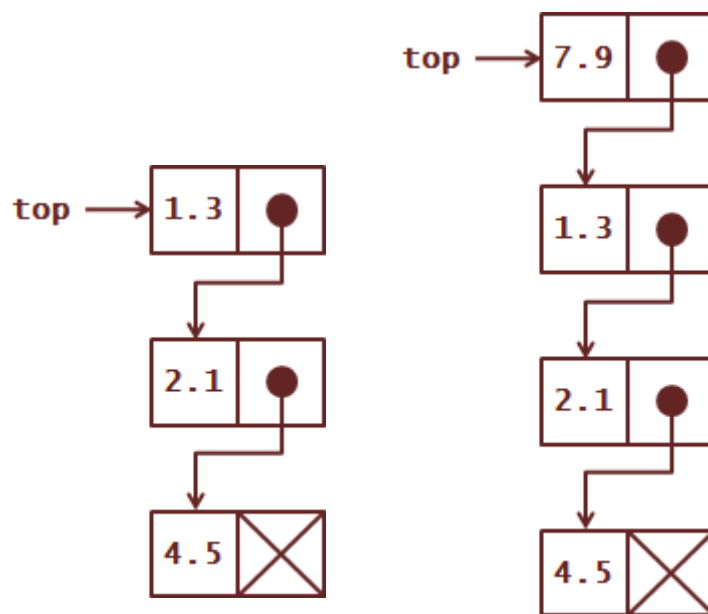
Problem 3 : Implentasi Fungsi Push

Definisi Masalah

Bentuk prototype dari fungsi push() umumnya adalah sebagai berikut ini:

```
Void push(double elmt , stacktype *stack);
```

Fungsi push digunakan untuk menambah element ke bagian teratas dari stack. Misalkan fungsi ini dipanggil dengan perintah **push(7.9, &stack)** pada saat kondisi pada gambar 2, maka fungsi akan berjalan sebagai berikut:



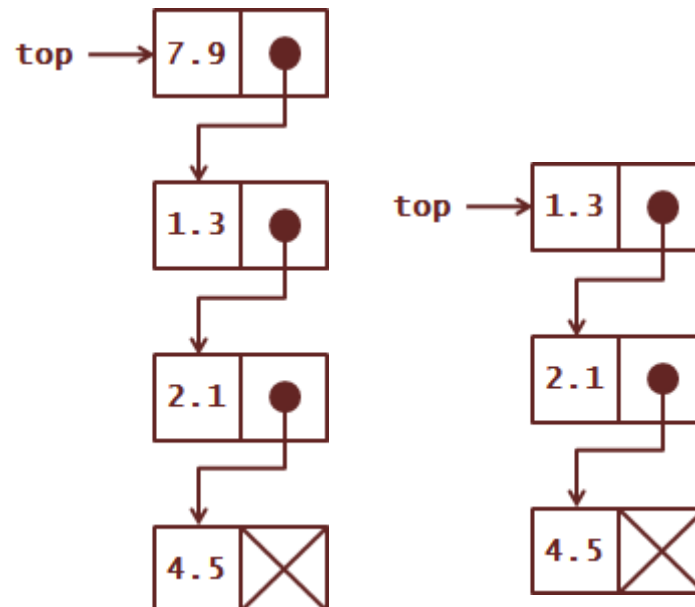
Gambar 4: Sebelum dan setelah push(7.9, &stack) dieksekusi

Implementasikanlah fungsi push() ini, hingga bekerja dengan baik.

Problem 4 : Implementasi Fungsi Pop

Definisi Masalah

Fungsi pop adalah kebalikan dari fungsi push, berguna untuk menghilangkan element teratas dari stack.



Gambar 5: Sebelum dan sesudah pop(&stack) dieksekusi

Ada perbedaanya dengan fungsi push. Pada fungsi push kita tidak perlu mengembalikan nilai hasil dari proses fungsi. Pada fungsi pop dilakukan pengembalian atas nilai atau data yang kita ambil. Pada contoh gambar 5, fungsi pop akan mengembalikan nilai 7.9.

Kode prototype dari fungsi pop adalah

```
Void pop(double stacktype *stack);
```

Implementasikanlah fungsi push() ini, hingga bekerja dengan baik.

Dari keseluruhan tugas diatas, buatlah program sederhana untuk menggunakan stack dengan jumlah maksimal stack adalah 5.

Deliverable

Buatlah keempat tugas diatas dalam satu program, disimpan dalam bentuk filetpe .c. simpan file dengan nama TP_Problem1234.c. Jangan lupa memberikan identitas (*header file*) di awal file ini. Pastikan program dapat dikompilasi dan dijalankan dengan benar.

File yang akan dikirim akan memiliki susunan kurang lebih seperti di bawah ini:

```
/* ... library, definition, etc */  
  
typedef struct node  
{
```

```

    /** isi apa yang kurang
    Pada struktur ini
    ... */
} node;

typedef struct stacktype
{
    /** isi apa yang kurang
    Pada struktur ini
    ... */
} stacktype;

/** prototype fungsi */
void push(double elmt, stacktype *stack);
double pop(stacktype *stack);

int main(void)
{
    /** fungsi utama */
}

/** implementasi fungsi */
void push(double elmt, stacktype *stack)
{
    /** implementasi fungsi push */
}

double pop(stacktype *stack)
{
    /** implementasi fungsi pop */
}

```

Petunjuk Penyerahan Tugas Pendahuluan Modul X

Simpan file (TP_Problem1234.c) dalam satu folder. Gunakan program WinRAR untuk mengompresi menjadi arsip .Zip. Penamaan file Zip menggunakan format yang telah diberikan. File .Zip ini yang akan di-submit melalui Google Classroom Praktikum PMC 2019. Hanya file kode saja yang dimasukkan ke dalam arsip .Zip. File *executable* tidak perlu dimasukkan.

Selesai