

Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 7 adalah **Senin, 28 Oktober 2024 pukul 06.00 WIB**.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan: **TP_MOD_[XX]_NIM_NAMA.pdf**
9. **SOAL TEORI WAJIB DIKERJAKAN TULIS TANGAN, TIDAK BOLEH DIKETIK!**

Contoh:

```
int searchNode_130190XXXX (List L, int X);
```

CP (WA):

- Raihan (089638482851)
- Kayyisa (085105303555)
- Abiya (082127180662)
- Rio (081210978384)

SELAMAT MENERJAKAN^^

TUGAS PENDAHULUAN MODUL 7 STACK

Anda diminta untuk membuat stack dengan representasi statis, yaitu menggunakan array. Untuk itu buatlah ADT stack pada file “stack.h” dengan struktur berikut:

Notasi Algoritmik	Bahasa C++
<pre> type infotype : char type stack : < info : array[1..15] of infotype top : integer > </pre>	<pre> typedef int infotype; struct stack { infotype info[15]; int Top; }; </pre>

Ilustrasi untuk struktur di atas adalah:

Info

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Top

Elemen info[1..15] akan digunakan untuk menampung data char. Sedangkan Top akan digunakan untuk menampung informasi mengenai indeks elemen info yang paling atas. Jadi, Top = 0 artinya stack dalam keadaan kosong. Top = 1, artinya stack berisi satu elemen.

Prosedur dan fungsi yang digunakan adalah

Notasi Algoritmik	Bahasa C++
<pre> procedure createStack(input/output S : stack) { I.S. sembarang F.S. terbentuk stack dengan Top = 0 } </pre>	<pre> void createStack(stack &S); </pre>
<pre> function isEmpty(S : stack): boolean { Mengembalikan nilai true jika stack kosong } </pre>	<pre> bool isEmpty(stack S); </pre>
<pre> function isFull(S : stack): boolean { Mengembalikan nilai true jika stack penuh } </pre>	<pre> bool isFull(stack S); </pre>

<u>procedure</u> push(<u>input/output</u> S : stack, <u>input</u> x : infotype) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 }	<u>void</u> push(stack &S, infotype x);
<u>function</u> pop(<u>input/output</u> S : stack): infotype { Mengembalikan nilai pada indeks Top, Top = Top - 1 }	<u>int</u> pop(stack &S);
<u>procedure</u> printInfo(<u>input</u> S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack }	<u>void</u> printInfo(stack S);

Gunakan alias sebagai berikut:

Top(S) untuk (S).Top

info(S) untuk (S).info

Buatlah implementasi ADT stack pada file "stack.cpp".

Notasi Algoritmik	Bahasa C++
<u>procedure</u> createStack(<u>input/output</u> S : stack) { I.S. sembarang F.S. terbentuk stack dengan Top = 0 } Kamus: { Tidak ada kamus } Algoritma: Top(S) ← 0	<u>void</u> createStack(stack &S) { /* Lengkapi kodenya */ }

Notasi Algoritmik	Bahasa C++
<u>function</u> isEmpty(S : stack): <u>boolean</u> { Mengembalikan nilai true jika stackkosong } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 0 <u>then</u> → <u>true</u> <u>else</u> → <u>false</u> { <u>end if</u> }	<u>bool</u> isEmpty(stack S) { /* Lengkapi kodenya */ }

Notasi Algoritmik	Bahasa C++
<p><u>function</u> isFull(S : stack): <u>boolean</u> { Mengembalikan nilai true jika stack penuh } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 15 <u>then</u> → <u>true</u> <u>else</u> → <u>false</u> { end if }</p>	<pre>bool isFull(stack S) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p><u>procedure</u> push(input/output S : stack, <u>input</u> x : <u>infotype</u>) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 } Kamus: { Tidak ada kamus } Algoritma: if isFull(S) = <u>false then</u> Top(S) ← Top(S) + 1 Info(S)[Top(S)] ← x { end if }</p>	<pre>void push(stack &S, infotype x) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p><u>function</u> pop(input/output S : stack): infotype { Mengembalikan nilai infotype yang ada pada indeks Top, Top = Top - 1 } Kamus: x : infotype Algoritma: x ← info(S)[Top(S)] Top(S) ← Top(S) - 1 → x</p>	<pre>int pop(stack &S) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<pre> procedure printInfo(input S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack } Kamus: i : integer Algoritma: for i ← Top(S) downto 1 do output(info(S)[i]) { end for </pre>	<pre> void printInfo(stack S) { /* Lengkapi kodenya */ } </pre>

Untuk menguji implementasi stack buatlah kode pada "main.cpp" sehingga menghasilkan output berikut ini dengan memanggil function/procedure yang sudah dipakai di atas:

- Digit terakhir NIM MOD 4 sisa 0 :

Output:

IFLABJAYA
JAYA

- Digit terakhir NIM MOD 4 sisa 1 :

Output:

HALOBANDUNG
BANDUNG

- Digit terakhir NIM MOD 4 sisa 2 :

Output:

PERCAYADIRI
DIRI

- Digit terakhir NIM MOD 4 sisa 3 :

Output:

STRUKTURDATA
DATA

Note:

Output:

isi stack awal

isi stack sesudah pop

Semoga Selalu diberi kemudahan^^