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 MENGERJAKAN = PENGURANGAN POIN JURNAL / TES ASESMEN.
4. Deadline pengumpulan TP Modul 7 adalah Senin, 28 Oktober 2024 pukul 06.00 WIB.

## TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.

1. DILARANG PLAGIAT (PLAGIAT = E).
2. Kerjakan TP dengan jelas agar dapat dimengerti.
3. File diupload di LMS menggunakan format PDF dengan ketentuan: TP\_MOD\_[XX]\_NIM\_NAMA.pdf
4. **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 MENGERJAKAN^^**

**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++ |
| type infotype : char | **typedef int** infotype; |
| type stack : < | **struct** stack { |
| info : array[1..15] of infotype | infotype info[15]; |
| top : integer  > | **int** Top;  }; |

Ilustrasi untuk struktur di atas adalah:



Info

Top

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

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

|  |  |
| --- | --- |
| procedure push(input/output S : stack, input x : infotype)  { I.S. mungkin kosong  F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1} | **void** push(stack &S, infotype x); |
| function pop(input/output S : stack): infotype  { Mengembalikan nilai pada indeks Top, Top = Top - 1 } | **int** pop(stack &S); |
| procedure printInfo(input S : stack)  { I.S. stack mungkin kosong  F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack } | **void** 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++ |
| procedure createStack(input/output S : stack)  { I.S. sembarang  F.S. terbentuk stack dengan Top = 0 } Kamus:  { Tidak ada kamus } Algoritma:  Top(S) ← 0 | **void** createStack(stack &S) {  /\* Lengkapi kodenya \*/  } |

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

|  |  |
| --- | --- |
| Notasi Algoritmik | Bahasa C++ |
| function isFull(S : stack): boolean  { Mengembalikan nilai true jika stackpenuh } Kamus:  { Tidak ada kamus } Algoritma:  if Top(S) = 15 then  → true  else  → false  { end if} | **bool** isFull(stack S) {  /\* Lengkapi kodenya \*/  } |

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

|  |  |
| --- | --- |
| Notasi  Algoritmik | Bahasa  C++ |

|  |  |
| --- | --- |
| function 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 | **int** pop(stack &S) {  /\* Lengkapi kodenya \*/  } |

|  |  |
| --- | --- |
| Notasi  Algoritmik | Bahasa  C++ |
| 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 } | **void** printInfo(stack S) {  /\* Lengkapi kodenya \*/  } |

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:

## I F L A B J A Y A J A Y A

* Digit terakhir NIM MOD 4 sisa 1 :

# Output:

## H A L O B A N D U N G B A N D U N G

* Digit terakhir NIM MOD 4 sisa 2 :

# Output:

## P E R C A Y A D I R I D I R I

* Digit terakhir NIM MOD 4 sisa 3 :

Note:

# Output:

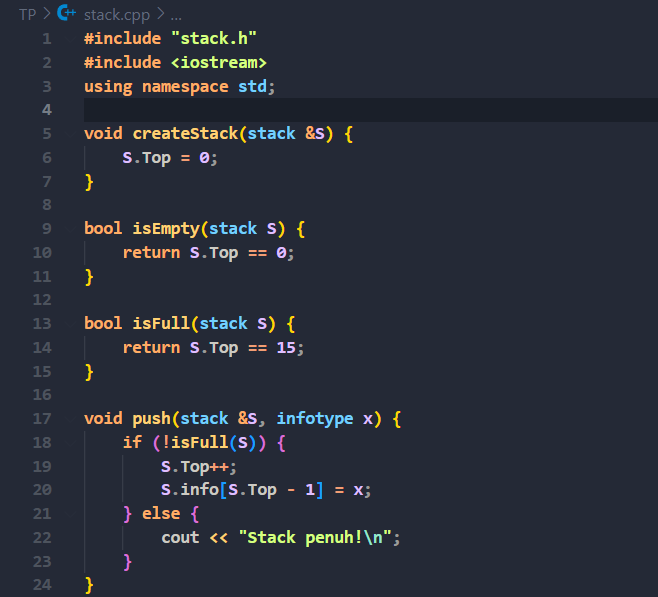
**Output:**

## S T R U K T U R D A T A D A T A

*isi stack awal*

*isi stack sesudah pop*

1. *Stackcpp*

**

*A screen shot of a computer program

Description automatically generated*

1. *Stackh*

*A screen shot of a computer program

Description automatically generated*

1. *Maincpp*

*A screen shot of a computer program

Description automatically generated*

*A screen shot of a computer program

Description automatically generated*

*A screen shot of a computer

Description automatically generated*

**Semoga Selalu diberi kemudahan^^**