

LAPORAN PRAKTIKUM
PERTEMUAN 7
STACK



Nama :

Martryatus Sofia 2311104003

Dosen :

Yudha Islami Sulistya, S.Kom., M.Cs

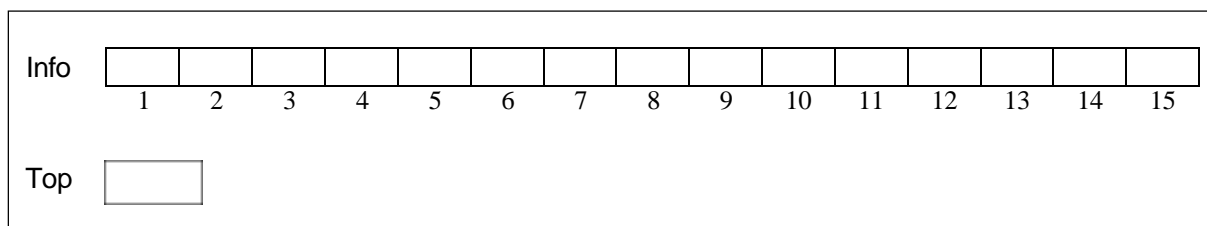
PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO 2024

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:



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

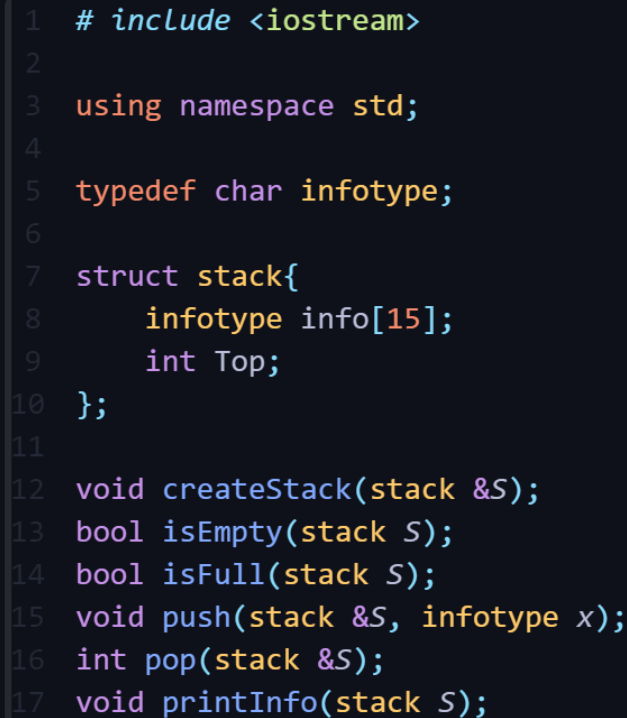
Note:

Output:

isi stack awal

isi stack sesudah pop

Stack.h



```
1  #include <iostream>
2
3  using namespace std;
4
5  typedef char infotype;
6
7  struct stack{
8      infotype info[15];
9      int Top;
10 };
11
12 void createStack(stack &S);
13 bool isEmpty(stack S);
14 bool isFull(stack S);
15 void push(stack &S, infotype x);
16 int pop(stack &S);
17 void printInfo(stack S);
```

typedef char infotype: Mendefinisikan infotype sebagai tipe data char, yang artinya stack ini akan menyimpan data berjenis karakter.

void createStack(stack &S): Fungsi createStack digunakan untuk menginisialisasi stack agar bisa digunakan.

bool isEmpty(stack S): Fungsi isEmpty digunakan untuk memeriksa apakah stack dalam keadaan kosong. Biasanya, stack dianggap kosong jika Top berada di nilai awal (misalnya -1).

bool isFull(stack S) : Fungsi isFull digunakan untuk memeriksa apakah stack sudah penuh, yaitu ketika Top mencapai indeks maksimum (dalam hal ini 14, karena ukuran array adalah 15).

void push(stack &S, infotype x) : Fungsi push untuk menambahkan elemen baru x ke dalam stack. Elemen ini ditambahkan di posisi paling atas, yaitu Top.

int pop(stack &S) : Fungsi pop digunakan untuk mengeluarkan atau menghapus elemen paling atas dari stack. Fungsi ini biasanya mengembalikan elemen yang dikeluarkan dari stack.

void printInfo(stack S) : Fungsi *printInfo* digunakan untuk menampilkan semua elemen yang ada di dalam stack mulai dari elemen paling bawah hingga paling atas.

Stack.cpp

```
1 #include "stack.h"
2 #include <iostream>
3 using namespace std;
4
5 // Inisialisasi stack kosong
6 void createStack(stack &S) {
7     S.Top = 0;
8 }
9
10 // Mengecek apakah stack kosong
11 bool isEmpty(stack S) {
12     return S.Top == 0;
13 }
14
15 // Mengecek apakah stack penuh
16 bool isFull(stack S) {
17     return S.Top == 15;
18 }
19
20 // Menambahkan elemen ke dalam stack
21 void push(stack &S, infotype x) {
22     if (!isFull(S)) {
23         S.Top++;
24         S.info[S.Top - 1] = x;
25     } else {
26         cout << "Stack penuh, tidak bisa menambah elemen.\n";
27     }
28 }
29
30 // Mengambil elemen dari stack
31 int pop (stack &S) {
32     if (!isEmpty(S)) {
33         int x = S.info[S.Top - 1];
34         S.Top--;
35         return x;
36     } else {
37         cout << "Stack kosong, tidak ada elemen untuk diambil.\n";
38         return '\0'; // Mengembalikan karakter null jika kosong
39     }
40 }
41
42 // Menampilkan semua elemen dalam stack
43 void printInfo(stack S) {
44     if (!isEmpty(S)) {
45         for (int i = S.Top - 1; i >= 0; i--) {
46             cout << S.info[i] << " ";
47         }
48         cout << endl;
49     } else {
50         cout << "Stack kosong.\n";
51     }
52 }
53 }
54
```

Main.cpp

```

1 #include "stack.h"
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     stack S;
7     createStack(S);
8
9     // Digit terakhir NIM MOD 4 sisa 3, maka output "S T R U K T U R D
10    A T A"
11    char data[] = {'S', 'T', 'R', 'U', 'K', 'T', 'U', 'R', ' ', 'D', 'A', 'T', 'A'};
12
13    // Push semua karakter ke dalam stack
14    for (char c : data) {
15        push(S, c);
16    }
17
18    // Tampilkan isi stack awal
19    cout << "Isi stack awal: ";
20    printInfo(S);
21
22    // Pop 8 kali untuk menyisakan "D A T A" saja
23    for (int i = 0; i < 8; i++) {
24        pop(S);
25    }
26
27    // Tampilkan isi stack setelah pop
28    cout << "Isi stack setelah pop: ";
29    printInfo(S);
30
31    return 0;
32 }

```

include "stack.h" : Ini mengimpor file header stack.h, yang berisi deklarasi struktur stack dan fungsi-fungsi terkait yang didefinisikan dalam kode sebelumnya (createStack, push, pop, dan printInfo).

int main() Fungsi utama main() adalah tempat program dieksekusi.

Membuat variabel stack S dan menginisialisasinya menggunakan createStack(S), sehingga stack siap digunakan. Membuat array karakter data yang berisi kata "S T R U K T U R D A T A". Data ini digunakan karena digit terakhir NIM MOD 4 memiliki sisa 3, sehingga teks yang dipilih adalah "S T R U K T U R D A T A". Melakukan perulangan untuk setiap karakter c dalam array data dan menambahkannya ke dalam stack S dengan fungsi push. Semua karakter dari array data dimasukkan ke dalam stack satu per satu. Menampilkan seluruh elemen dalam stack S setelah proses push selesai. Fungsi printInfo akan mencetak elemen-elemen dari yang paling bawah hingga yang paling atas. Melakukan operasi pop sebanyak 8 kali, sehingga 8 karakter teratas dihapus dari stack. Setelah ini, stack hanya akan menyisakan karakter "D A T A". Menampilkan isi stack S setelah operasi pop. Hanya karakter "D A T A" yang tersisa, yang ditampilkan oleh printInfo.

OUTPUT

```
collect2.exe: error: ld returned 1 exit status
PS F:\Kuliah\Semester3\Strukdat - praktikum\07_Stack\07_Stack\TP> g++ main.cpp stack.cpp -o program
PS F:\Kuliah\Semester3\Strukdat - praktikum\07_Stack\07_Stack\TP> ./program
Isi stack awal: A T A D   R U T K U R T S
Isi stack setelah pop: K U R T S
```

Semoga Selalu diberi kemudahan^^